OBJECT-ORIENTED MODEL FOR INDUCTANCE COMPUTATION OF A LINEAR SRM

S.J. Watkins, J. Čorda
School of Electronic and Electrical Engineering
The University of Leeds
Woodhouse Lane, Leeds, LS2 9JT, UK
Tel: +44 113 3532042
Email: eensjwa@leeds.ac.uk

Abstract—**The paper presents a method of computing the minimum and maximum inductance of a cylindrical linear switched reluctance motor based on an object-oriented model. This model is a useful tool for rapid computation of magnetic characteristics of the motor during the process of design optimisation. The key feature of the method is that the machine is decomposed into simple 3D segments with analytically derived equations, which are merged by applying an object-oriented programming approach to obtain the values of phase inductances for the extreme positions.**

Key words— **linear switched reluctance motor, inductance modelling, object-oriented programming.**

1. INTRODUCTION

The linear cylindrical switched reluctance motor (LSRM) [1] offers advantages through its very simple construction without permanent magnets. A longitudinal cross-section of the four-phase motor is shown in Figure 1. Each phase comprises a single circular band winding inside a back-iron ring, sandwiched between two stator pole discs. The phases are magnetically self-contained and separated from each other by non-magnetic spacing rings. The inner part, considered here as the mover, has simple saliencies produced by transverse slots and allows long-stoke operation of the machine either as an open-loop stepper or as a switched reluctance linear motor with mover position feedback commutation.

To aid the identification of the optimum machine design for a target thrust-speed characteristic, it is beneficial to have a rapid computer model of the machines minimum and maximum inductances. Previous work [2] has derived analytical equations for achieving this task. This paper presents a technique that decomposes the LSRM into a set of standard 3D shapes which enables modelling using object-oriented programming (OOP).

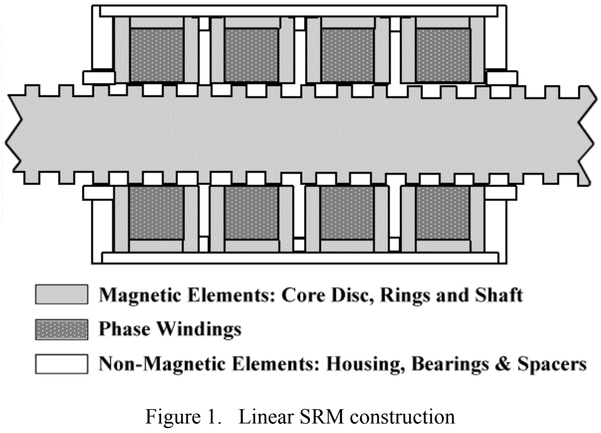
Magnetic Elements: Core Disc, Rings and Shaft
Phase Windings
Non-Magnetic Elements: Housing, Bearings & Spacers

Figure 1. Linear SRM construction

2. OBJECT-ORIENTED PROGRAMMING

2.1. Key Features Overview

In the real world any object can be thought of as having characteristics, or states, and behaviour. Object-oriented programming uses this principle to instantiate objects within the software, whereby each object has variables (states) and functions or methods (behaviour). In C++ [3], the class is the blueprint for an object and contains the necessary code to define the required object. This approach helps with modularity in terms of maintaining software, and any modifications can take place at the class level. Different objects which share common traits can be instantiated through the use of derived subclasses. This is achieved by defining an abstract class with generic states and behaviour and then through inheritance defining subclasses which have modifications for the actual objects required. This allows efficient software reuse and behaviour enhancements to be done at the abstract class level. Another important feature of C++ is the operator expression which can be overloaded, i.e. its behaviour can be modified in the abstract class or subclasses. Arithmetic operators can be defined for objects, which give specific behaviour to their use in order to make code clearer.

Applying this to inductance modelling, an abstract class is a 3D shape describing a segment of a flux path and contains methods for the standard magnetic circuit equations (1) and (2) for reluctance $\Re$, permeance P and mmf F. The class code variables are: permeability μ, path length l and cross-sectional area A.

$$\Re = 1/P = \frac{l}{\mu A} \tag{1}$$

$$F = \Re \, \phi \tag{2}$$

where ϕ denotes the flux.

The permeability, as a variable, is itself defined as a class, allowing the representation of non-linear ferromagnetic materials by B-H curve data, or of the air paths by the constant value μ_o. Class operator functions such as '+' for series connection, '/' for parallel connection and '=' for assignment, allow new 3D objects to be instantiated in the code which represents combined lumped flux path elements. Once the complete magnetic circuit has been instantiated in software as a combination of flux path shapes, the inductance can be found at a particular current as

$$L = \psi / I \quad \text{or} \quad L = N^2 / \Re_{tot} \tag{3}$$

where ψ denotes the flux-linkage.

2.2. Flux-Shape Classes in the LSRM

Using the abstract class *Segment* as a base, a set of specific 3D shape classes can be defined representing the flux path regions within the LSRM. In the cylindrical machine, the flux paths can be divided up into two basic shapes, as illustrated by Figure 2 related to the phase core. The *Cylinder* segment for straight-line flux paths and the *Torus* segment for circular arc flux paths. Two additional segments, *WindingCylinder* and *WindingTorus*, are also defined for similarly shaped regions incorporating the winding.

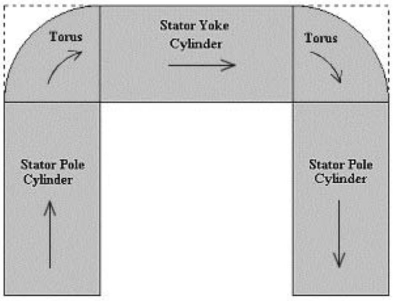

Figure 2. Flux paths in the Stator Phase Core

Cylinder class has three physical (dimensional) properties – width w, inner diameter d_i, and outer diameter d_o , as shown in Figure 3. When evaluating the individual mmf contribution of each motor segment, the material permeability characteristic and flux path direction must also be known. For instance, the flux in the stator yoke and the mover shaft passes in axial direction, while in the rest of the circuit the flux passes in radial direction. Computation of the path length l and cross-sectional area A will therefore be specific with regard to the flux direction.

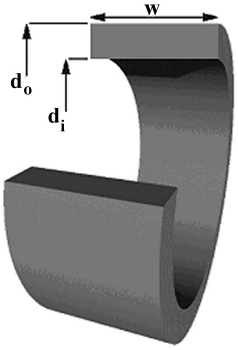

Figure 3. Cylinder segment object

The *Cylinder* class has different 'methods' for calculating the magnetic flux path dimensions for axial flux direction given by (4) and (5), and for the radial flux direction given by (6) and (7). (The area A in the radial flux direction is the mean value for the segment.)

$$A = \pi(d_o^2 - d_i^2)/4 \tag{4}$$

$$l = w \tag{5}$$

$$A = w\pi(d_o + d_i)/2 \tag{6}$$

$$l = (d_o - d_i)/2 \tag{7}$$

Torus class represents a three-dimensional region where flux passes in a circular arc path, as shown in Figure 4. Its dimensional properties – inner radius r_i, outer radius r_o and azimuthal diameter d, are used to compute the class member properties – the mean path length l and mean cross-sectional area A of the lumped magnetic circuit element.

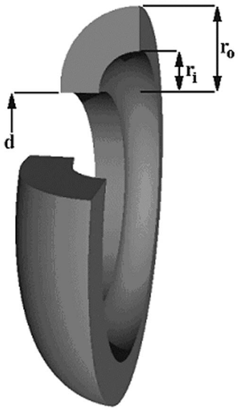

Figure 4. Torus segment object (outer flux path version)

The terms A and l for the *Torus* segment, shown in Figure 4, where the flux path is in one of the outermost quarter parts of a complete ring, which is centred on the machine's mover axis, are given analytically by (8) and (9). The terms A and l for the innermost flux path *Torus* object are given by (10) and (11).

$$A = \pi\Big((r_o - r_i)d + (r_o^{\,2} - r_i^2)/\sqrt{2}\Big) \tag{8}$$

$$l = A\Big(2d\ln(r_o/r_i) + 4(r_o - r_i)/\sqrt{2}\Big)^{-1} \tag{9}$$

$$A = \pi\Big((r_o - r_i)d - (r_o^{\,2} - r_i^2)/\sqrt{2}\Big) \tag{10}$$

$$l = A\Big(2d\ln(r_o/r_i) - 4(r_o - r_i)/\sqrt{2}\Big)^{-1} \tag{11}$$

WindingCylinder class represents a *Cylinder* within a winding but with the average cross-sectional area modified due to the fact that the flux passing axially is not linked by all the winding turns. This fact is incorporated into the equation defining the magnetic field strength and is used in the flux-linkage equation to obtain an expression for the mean ratio *A/l*. The path length l in the axial direction is constant at all diameters and is equal to the cylinder width w, and the mean cross-sectional area A is given by:

$$A = \pi\Big(d_i(d_o - d_i)/6 + (d_o - d_i)^2/24\Big) \tag{12}$$

WindingTorus class represents the flux through the winding which has a circular arc path form. The path length and cross-sectional area are derived analytically by modifying the flux-linkage equation to take into account the reduced number of turns linking the flux. The term A is found in the same way as the *Torus* class given by (8) and (10), and the mean path length is then given by:

$$l = A\left(2\left(\frac{d}{2}\ln\left|\frac{r_o}{r_i}\right| + \frac{(r_o - r_i)}{\sqrt{2}} - \frac{d\pi(r_o^2 - r_i^2)}{8A_w} - \frac{\pi(r_o^3 - r_i^3)}{6\sqrt{2}A_w^2} + \frac{d\pi^2(r_o^4 - r_i^4)}{128A_w} + \frac{\pi^2(r_o^5 - r_i^5)}{80\sqrt{2}A_w^2}\right)\right)^{-1} \tag{13}$$

3. Inductance modelling

3.1. Machine 'Aligned Position' Class

The complete magnetic structure for the LSRM in the fully aligned position (Figure 5) can be modelled as a set of *Cylinder* shape objects, made of either mild steel or air. In the computer model, *Stator* and *Mover* objects are composed of their constituent *Cylinder* parts, with the poles and teeth defined as arrays of *Cylinders*. The stator poles extend only as far as the yoke inner diameter, while the width of stator yoke is same as that of the winding. To account for the change in the flux direction from axial through the yoke to radial in the poles, a *Torus* object is used in series with the yoke and each stator pole. No adjustment is made for the mover shaft/tooth boundary since the shaft cross-sectional area is far larger than the mover tooth root area, i.e. saturation is not likely to occur within the shaft. Further analytical investigation may reveal a more suitable method of modelling this juncture using the object-oriented approach.

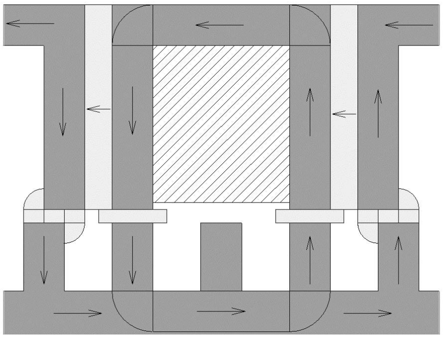

Figure 5. 'Aligned' approximated flux paths (inner phase)

In the radial direction the actual flux density increases towards the centre of the actuator and so simply modelling the stator poles and mover teeth as a single complete *Cylinder* segment may lead to computational errors, especially at higher saturating flux density levels. The object-oriented modelling approach allows a further subdividing of the stator poles and mover teeth into an array of *Cylinder* segments. This is done in software section related to the *Mover* and *Stator* classes.

The complete magnetic circuit object, *Aligned*, is composed of a *Stator* object, a *Mover* object and two *Cylinder* objects representing the air-gaps. The individual component mmfs are added together to give the total mmf for a given flux ϕ. To take into account the fringing effects in the air-gap, the air-gap object width can be assigned to be larger than the pole width by a factor that includes Carter's coefficient. The Carter's coefficient σ and effective air-gap width [2] are given by:

$$w_F = (w + 2r)\sigma - 2r = w - (2r(1-\sigma)) \tag{14}$$

$$r = u/\pi \tag{15}$$

$$\sigma = \frac{2}{\pi}\left(\arctan(\frac{2r}{g}) - \frac{g}{4r}\ln(1 + (\frac{2r}{g})^2)\right) \tag{16}$$

where u is the length of inter-phase spacing, and g is the radial air-gap length.

3.2. Machine 'Unaligned Position' Class

This class represents the unaligned magnetic circuit for LSRM's inner and outer phases. Figure 6 illustrates the approximated flux paths used in the model when one of the two inner phase windings is energised.

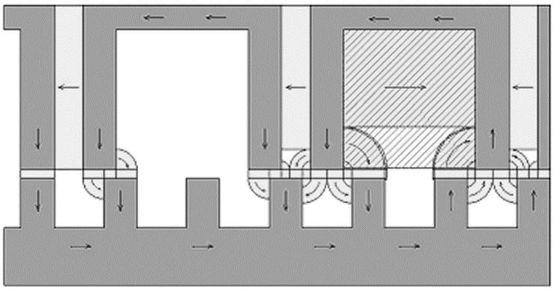

Figure 6. 'Unaligned' approximated flux paths (inner phase)

When the stator poles of excited phase are in full misalignment with the mover teeth, the mmf component in the iron is insignificant in comparison with the component in the air. This allows for the iron to be considered as approximately infinitely permeable. The real flux paths have therefore been approximated by either a *Cylinder* or *Torus* shape and adhere to the basic premise that the flux enters and exits an iron core part at right-angles to the iron's boundary. Flux paths within the phase core take into account the distribution of the winding and use the *WindingCylinder* and *WindingTorus* classes for computing the permeance.

The point at which the inner phase flux path diverges is defined by the height h above the stator pole tip, and this is determined by equating the lengths of boundary lines of along the paths from the diverging point, i.e.

$$h = \left(\frac{2+\pi}{2\pi}\right)u \tag{17}$$

where u is the length of inter-phase spacing.

The point at which the flux diverges inside the winding is defined by the height k above the stator pole tip, and this is calculated using the same reasoning as applied to h. The following equations relates k to the LSRM's physical dimensions:

$$k = \frac{w - 2g + \pi(m+t)}{2\pi} \tag{18}$$

where

$$m = (s - v)/2 \tag{19}$$

s is the distance between mover teeth
t is the mover tooth width
v is the stator pole width
g is the radial air-gap length
w is the winding width.

The difference between the inner phase and outer phase are dealt with through the asymmetry introduced by the adjacent phase flux paths. The leakage flux passing between the adjacent phase poles returns via the stator pole tips and the mover. For the inner phase, the leakage flux paths are symmetrical through both adjacent phases with an additional flux path added for the phase on the outside of the machine. The outer phase has one leakage flux path through its adjacent inner phase, with two additional flux paths for the other two phases. To complete the approximated flux path pattern for the whole machine, *Torus* object leakage flux paths are assigned for flux passing from the machine's outer pole sides to the mover teeth.

4. MODEL VALIDATION

The model was validated against an existing prototype machine which has the following dimensions:

s = 6mm (mover slot width)
t = 4mm (mover tooth width)
u = 3.5mm (inter-phase spacing)
v = 4mm (stator pole width)
w = 16mm (stator yoke width)
d_o = 80mm (machine outside diameter)
d_i = 74mm (stator yoke inside diameter)
d_m = 40mm (mover pole diameter)
g = 0.2mm (air-gap length)
d_{sh} = 28mm (mover shaft diameter)

Table 1 contains measured and computed unsaturated inductance values (averaged up to 300 Ampere-turns) for the prototype machine.

TABLE 1 MEASURED AND COMPUTED INDUCTANCE

	L_{max} (inner)	L_{max} (outer)	L_{min} (inner)	L_{min} (outer)
Measured	234 mH	210 mH	125 mH	92.3 mH
Computed	239 mH	206 mH	114 mH	80.0 mH

The discrepancy between computed and measured values is more noticeable in the unaligned position (minimum inductance). This is principally due to the fact that the flux paths in the aligned position are more constrained (guided by iron core) which enables somewhat better approximations to be used in estimating the maximum inductance.

The magnetic characteristics are shown in Figure 7. At higher currents the iron parts of the machine become saturated but the percentage discrepancies between the computed and measured results are not significantly changed compared to an unsaturated state. This indicates that the model accounts adequately for the effect of magnetic saturation.

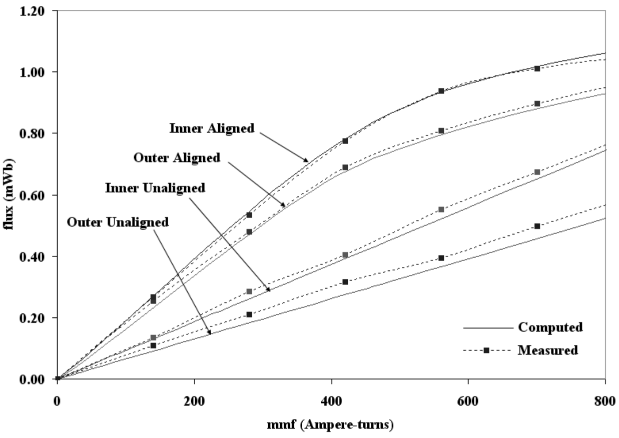

Figure 7. Calculated and measured magnetic characteristics

5. CONCLUSIONS

The object-oriented approach applied in computation of inductances of complex magnetic circuit such as the cylindrical LSRM allows the inclusion of flux paths for all machine's phases in a more detailed way compared to the analytical method originally employed. The use of basic abstract three-dimensional magnetic circuit elements, to model the machine's inductances in the extreme positions, avoids errors which inevitably creep-in when dealing with complex algebraic equations. A comparison of the computed results with those obtained by measurements on the prototype LSRM, show a maximum discrepancy of 13% for the minimum inductance and 5% for the maximum inductance. This compares with 16% and 10%, when the original analytical method is applied. The improvement in accuracy stems from the inclusion of more detailed flux paths in the inductance computations by the object-oriented method.

REFERENCES

[1] J. Čorda and E. Skopljak, "Linear switched reluctance actuator", Proceedings of EMD '93, the 6th IEE International Conference on Electrical Machines and Drives, IEE Conference Publication No. 376, Oxford, UK, 8 – 10 September, 1993, pp 535 - 539.

[2] J. Čorda, "Prediction of static characteristics of cylindrical linear variable-reluctance actuator", Proceedings of LDIA '98, the 2nd International Symposium on Linear Drives for Industrial Applications, Tokyo, Japan, 8 - 10 April, 1998, pp 367 - 370.

[3] B. Stroustrop, "The C++ programming language", Addison Wesley. ISBN: 0201539926, January, 1991.