

Article

Absolutely Feasible Synchronous Reluctance Machine Rotor Barrier Topologies with Minimal Parametric Complexity

Branko Ban ^{1,2,*}  and Stjepan Stipetic ¹ 
¹ Department of Electric Machines, Drives and Automation, Faculty of Electrical Engineering and Computing (FER), University of Zagreb, 10000 Zagreb, Croatia; stjepan.stipetic@fer.hr

² Torquery Consulting, 41830 Göteborg, Sweden

* Correspondence: branko.ban@fer.hr

Abstract: The first step in the synchronous reluctance machine design is the selection of rotor flux barrier type. The literature provides various barrier construction methods with a common goal of reducing parametric complexity. However, too excessive simplification can lead to decreased performance, while overly complex geometries tend to increase optimization time. This paper presents a set of novel flux barrier construction methods with an increased degree of freedom and minimal geometrical complexity. The paper proposes four topologies based on circular, hyperbolic, and original Zhukovsky lines. When considering parametrization complexity, the original Zhukovsky type is the simplest, but it has barrier depth limitations. Other topologies have equal complexity. The paper proposes a novel Modified Zhukovsky variable depth type based on geometrical conformal mapping of the original Zhukovsky lines. The step-by-step construction of each topology is presented in a form of pseudo-code with detailed comments and illustrations. Overall, the presented research provides a valuable starting point for the designer who wants to investigate different smooth rotor barrier topologies.

Keywords: synchronous reluctance; barrier comparison; rotor topology; conformal mapping; optimization



Citation: Ban, B.; Stipetic, S.

Absolutely Feasible Synchronous Reluctance Machine Rotor Barrier Topologies with Minimal Parametric Complexity. *Machines* **2022**, *10*, 206. <https://doi.org/10.3390/machines10030206>

Academic Editor: Toomas Vaimann

Received: 31 January 2022

Accepted: 10 March 2022

Published: 11 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, the synchronous reluctance machine (SyRM) has become commercially viable as a high-efficiency alternative to induction (IM) and interior permanent magnet machines (IPM). The main benefit of the SyRM is that the rotor has no squirrel cage, windings or magnets, just electric steel plates forming a rotor package.

Due to the highest torque and power density, interior rare earth permanent magnet synchronous machines (IPM) are preferred for automotive traction. On the other hand, the use of rare earth permanent magnet (PM) materials, such as neodymium or dysprosium, has historically been a commercial risk [1,2].

Currently, there is no commercial use of synchronous reluctance machines (SyRM) for automotive traction, but having in mind the market uncertainties and potential production cost reduction, they represent a possible alternative [3,4].

On the other hand, commercial vehicles apart from electric traction, have to actuate additional body systems (usually powered by some sort of hydraulic pump). The interface towards external systems is referred as electric power take-off (e-PTO). Considering that the e-PTO needs to be reliable, robust, and cheap, SyRM is the preferred alternative [5–7].

In recent decades, SyRM research has focused on improving the rotor barrier design, minimizing the torque ripple, and increasing the power factor [8]. Literature provides references to several barrier topologies: circular [9,10], hyperbolic [11,12], Zhukovsky fluid type [13,14], segmented, etc. Open-source SyRE project offers more details and instructions on geometry generation [15].

The common goal in barrier construction strategy is the reduction of parametric complexity. However, too simplified rotor topologies can lead to decreased performance, while

too complex geometries yield better performance, but also tend to increase optimization time (increase is proportional to the parametric complexity).

By merging both approaches, this paper presents a set of flux barrier construction methods with an increased degree of freedom and minimal parametric complexity, based on non-dimensional rotor parameters.

Considering that sharp edges tend to cause mechanical issues at high rotational speeds, only barrier topologies based on smooth analytical functions have been analyzed (circular, hyperbolic, and Zhukovsky).

By definition, Zhukovsky barrier lines are streamlines and cannot be analytically modified to secure variable barrier depth, in this case, conformal mapping is used to create a novel Modified Zhukovsky barrier type with variable depth lines. The following chapters will demonstrate automated barrier design and related pseudo-code for the following topologies:

1. Circular variable depth (CrVD), Figure 1b
2. Variable eccentricity hyperbolic (HyVE), Figure 1c
3. Original Zhukovsky (Zh), Figure 1d (red)
4. Modified Zhukovsky variable depth (MZhVD), Figure 1d (blue).

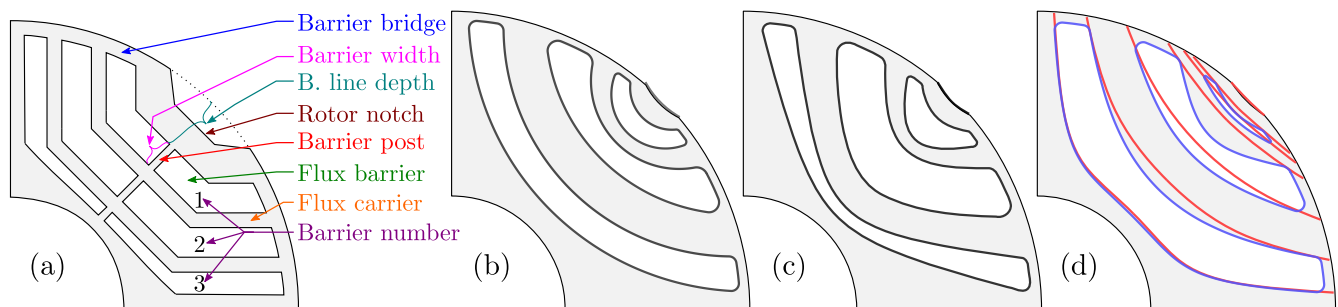


Figure 1. (a) SyRM terminology; SyRM rotor barrier types: (b) Circular variable depth; (c) Hyperbolic variable eccentricity; (d) Modified Zhukovsky (blue), original Zhukovsky (red).

2. Geometric Feasibility

The term feasibility usually refers to the solution and means that the solution satisfies all the given constraints. There is another type of feasibility called *geometric or model feasibility*. A geometrically feasible model is valid for solving if: there are no overlapping edges, negative lengths, or non-conventional geometric relations that will inevitably produce issues after optimization starts.

This is especially important when using template-based design software. The generation of such a non-valid model can be avoided in several ways (e.g., barrier 1 (blue) collides with barrier 2 (yellow), the collision is marked in red, Figure 2a). In the first case, the complete set of optimization parameters is initialized until geometric feasibility is achieved [16].

The alternative is forced feasibility, where each infeasible design is subjected to parameter modification until feasibility is reached, e.g., barrier 1 (blue) and barrier 2 (yellow) are modified until the minimum flux carrier width $w_{c \min}$ is reached, Figure 2b [7].

The final approach is to secure that the design is always feasible.

The example is a SyRM with three hyperbolic barriers where barrier depth is varied by hyperbolic eccentricity of each inner and outer barrier line (Figure 3b).

In some cases, the complete freedom in varying the eccentricity causes the inner and outer barrier collision (Figure 3a), which inevitably leads to infeasible designs.

On the other hand, if the designer wants to ensure feasibility, one solution is to parametrize the eccentricity of each outer barrier line relative to the eccentricity of each inner barrier, e.g., (1).

$$\begin{aligned}
 e_{3\text{in}} &\in [1.2 \ 1.3] & e_{3\text{out}} &\in [e_{3\text{in}} \ 1.35] \\
 e_{2\text{in}} &\in [e_{3\text{out}} \ 1.4] & e_{2\text{out}} &\in [e_{2\text{in}} \ 1.45] \\
 e_{1\text{in}} &\in [e_{2\text{out}} \ 1.5] & e_{1\text{out}} &\in [e_{1\text{in}} \ 1.55]
 \end{aligned} \tag{1}$$

Unfortunately, this option generates constantly changing parameter limits, which can lead to a suboptimal design. Considering the specifics of the SyRM rotor, the rotor geometry can be defined to always yield a feasible design thus achieving *absolute feasibility*. Instead of directly using eccentricities as parameters, a better approach is to use dimensionless inner and outer barrier depth parameters ($D_{\text{in}}, D_{\text{out}} \in [0, 1]$) for indirect calculation of respective eccentricities. The following text provides detailed instructions on absolutely feasible rotor construction with the corresponding pseudo-code which accepts any pole and barrier number.

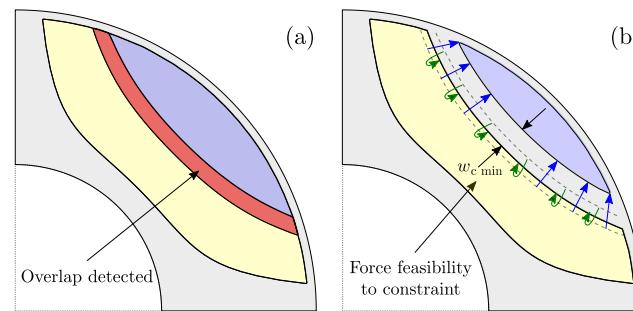


Figure 2. Infeasible geometry (a) and forced feasibility (b).

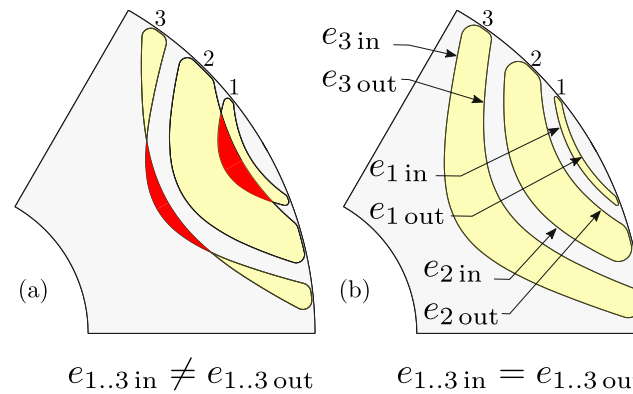


Figure 3. Illustration of infeasible (a) and feasible geometry (b).

3. Design Automation

The following figures are drawn for illustrative purposes and valid for a three barrier rotor ($k = 3$), naming and description of all parameters is explained in Table 1. Vector variables are bolded, e.g., \mathbf{R} is a variable vector, while R represents a scalar variable.

The initial step in rotor construction (Figure 4a) is to specify number of pole pairs (p), rotor barriers (k) and barrier bridge thickness (w_{bb}). The user then specifies dimensionless $\vartheta_{\text{min}}, \vartheta_{\text{max}} \in [0, 1]$ (Table 1, 18–19). Temporary construction points vector $\mathbf{E}_{1..k \text{ temp}}$ is then created with equidistant angular spacing $\Delta\vartheta_r = (\vartheta_{\text{max}} - \vartheta_{\text{min}})/k$. Barrier notch point (E_n) is defined via additional parameter ϑ_{notch} (Table 1, 20) relative to ϑ_{min} with radial component equal to rotor radius. Note that the entire geometry is initially constructed in vertical manner (center pole axis is at the angle of 90°).

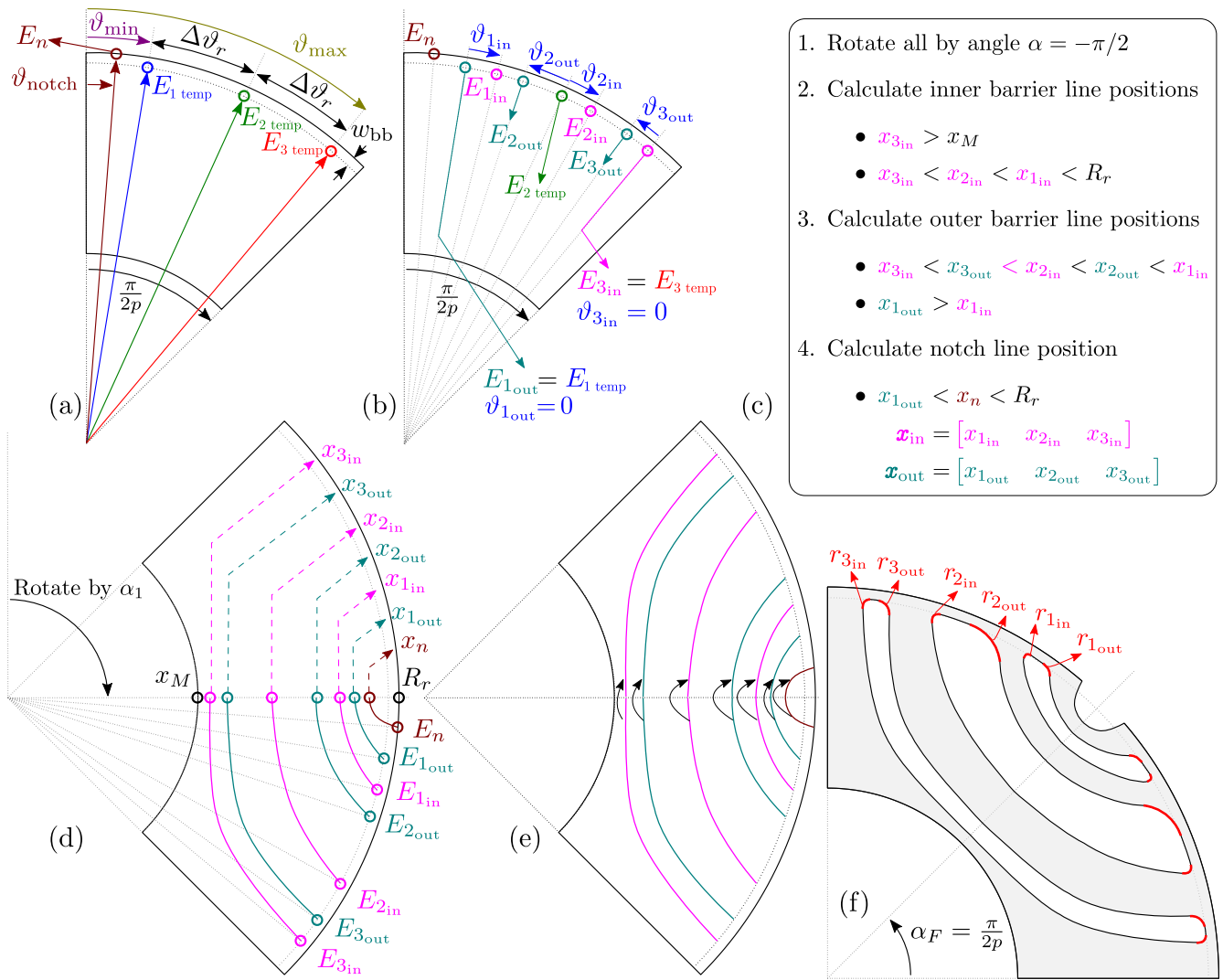


Figure 4. Rotor barrier construction procedure. Initial rotor construction step (a); Construction of inner and outer barrier line starting points (b); calculation of barrier intersection points (c); calculation of barrier vertices (d); vertex mirroring (e); rotation around center and addition of barrier fillets (f).

The second step (Figure 4b) is the construction of inner and outer barrier line starting points ($E_{\text{in}}, E_{\text{out}}$). The points are calculated relative to $E_{1..k \text{ temp}}$, based on additional set of dimensionless parameters $\vartheta_{1..k \text{ in}}, \vartheta_{1..k \text{ out}} \in [0, 1]$ (Table 1, 6–11).

The third step is rotation around the center point by the specified angle (in this case $\alpha_1 = -\pi/2$). Barrier line starting points $E_{\text{in}}, E_{\text{out}}, E_n$ (and additional depth parameters $D_{\text{in}}, D_{\text{out}}, D_n$ depending of the barrier type) are forwarded to the selected construction function which calculates barrier line intersection points ($x_{\text{in}}, x_{\text{out}}$). The intersection points are calculated based on depth parameters and feasibility conditions listed in Figure 4c. The function returns all inner and outer barrier line vertices ($X_{\text{in}}, Y_{\text{in}}, X_{\text{out}}, Y_{\text{out}}, X_n, Y_n$), Figure 4d.

The next step is mirroring line vertices around the horizontal axis (Figure 4e).

The final step is the rotation around the center point by the angle $\alpha_F = \pi/(2p)$. Barrier fillets ($r_{1..k \text{ in}}, r_{1..k \text{ out}} \in [0, 1]$, Table 1, 12–17) responsible for securing mechanical integrity of the rotor are added to the geometry, and final rotor geometry is exported as to the FEA tool (Figure 4f). Adding precise fillets to the discrete lines is a complex problem which is planned to be explained in the future publications. Detail barrier construction steps from Figure 4b–d are discussed in the following sections.

Table 1. Example list of design parameters. Color coding is according to Figure 4 variables.

No:	Description	Symbol	Value/Range	Unit
1	Rotor diameter	D_r	100	mm
2	Shaft diameter	D_{sh}	54	mm
3	Barrier number	k	3	-
4	Pole pairs	p	2	-
5	Barrier bridge	w_{bb}	0.3	mm
6	Point angle in	θ_{1in}	[0.2, 0.5]	-
7	Point angle out	θ_{1out}	0	-
8	Point angle in	θ_{2in}	[0, 0.3]	-
9	Point angle out	θ_{2out}	[0, 0.2]	-
10	Point angle in	θ_{3in}	0	-
11	Point angle out	θ_{3out}	[0, 0.5]	-
12–14	Corner rad. in	$r_{1..k_{in}}$	[0, 1]	-
15–17	Corner rad. out	$r_{1..k_{out}}$	[0, 1]	-
18	Min. angle	ϑ_{min}	[0.15, 0.3]	-
19	Max. angle	ϑ_{max}	[0.9, 0.95]	-
20	Notch angle	θ_n	[0.1, 1]	-
22–24	Barrier depths in	$D_{1..k_{in}}$	[0.2, 1]	-
25–27	Barrier depths out	$D_{1..k_{out}}$	[0.2, 1]	-
28	Notch depth	D_n	[0, 1]	-

Barrier Depth Variation

Considering that the barrier width has a substantial impact on the machine performance, this section will explain how inner and outer barrier depth coefficients affect each of the studied topologies, with a simplified presumption of equal line starting points.

Width of the each barrier depends on initial inner and outer line starting points E_{in} , E_{out} , and depth coefficients D_{in} , D_{out} . Depending on the depth parameter combination, barrier width can be variable, or uniform. Uniform width is a special case where CrVD barriers are concentric (Figure 5a, green). HyVE barriers can be approximately uniform when they have equal eccentricity (Figure 5b, green). These variants are included in CrVD and HyVE pseudo-code, and will not be studied in detail.

In general, CrVD and HyVE depth variation is unconstrained resulting in variable barrier width (Figure 5a,b, blue).

Zh barrier type (Figure 5c) is a special case because it does not support any depth variation. Barrier line depths are defined directly from starting points and cannot be modified. In order to explore the possible benefits of depth variation, Zh type has been modified as MZhVD where barrier depths have full freedom (Figure 5d).

An example of different barrier line depth parameter combinations is provided in the (Table 2).

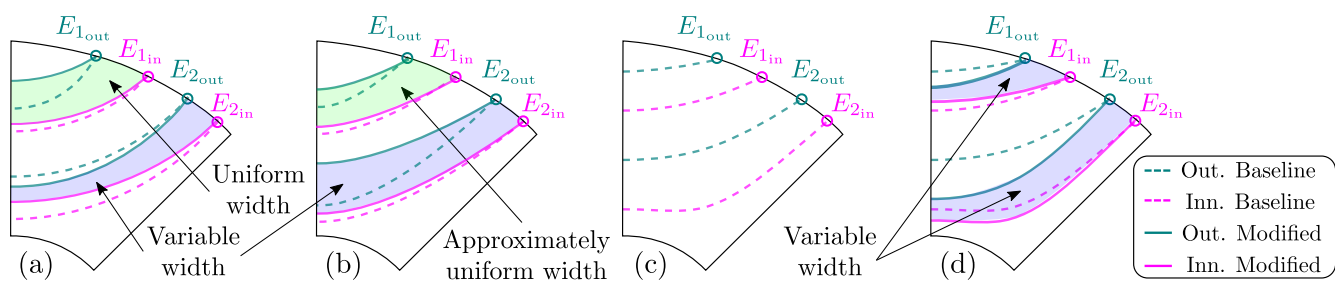


Figure 5. Barrier depth variation influence on different 2-barrier ($k = 2$) SyRM topologies. Circular (a); Hyperbolic (b), Zhukovsky (c) and Modified Zhukovsky barrier types (d).

Table 2. Illustrative depth coefficient table for Figure 5.

		Baseline Barrier Depths				Modified Barrier Depths			
	Abbr.	D_{1out}	D_{1in}	D_{2out}	D_{2in}	D_{1out}	D_{1in}	D_{2out}	D_{2in}
(a)	CrVD	0.35	0.50	0.65	0.90	0.40	0.55	0.70	0.80
(b)	HyVE	0.40	0.50	0.80	0.90	0.35	0.45	0.60	0.85
(c)	Zh	0.10	0.45	0.60	0.85	-	-	-	-
(d)	MZhVD	0.10	0.45	0.60	0.85	0.20	0.40	0.80	0.90

4. Standard Rotor Barriers

4.1. Zhukovsky Barrier Construction

After definition of all inner and outer barrier line starting points (E_{in}, E_{out}, E_n), Figure 6a, the entire geometry is rotated by $\alpha_{1Zh} = -(\pi/2 - \tau_{pole}/2)$ (Algorithm 1: In:2, Figure 6b, $\tau_{pole} = (2\pi)/(2p) = \pi/p$ is the angular pole step.). This partial rotation must be performed because Zhukovsky equations are defined on angular range $[0, \pi/p]$.

Algorithm 1 Construction of Zhukovsky barriers

```

1:  $\tau_{pole} = 2\pi/(2p)$ 
2: Rotate all by  $\alpha_{1Zh} = -(\pi/2 - \tau_{pole}/2)$  and get:  $E_{in}, E_{out}, E_n$ 
3:  $\Delta r = 0$ 
4:  $[X_{in}, Y_{in}, X_{out}, Y_{out}, X_n, Y_n] = \text{GETZHUKOVSKYLINES}(E_{in}, E_{out}, E_n)$ 

5: function GETZHUKOVSKYLINES( $E_{in}, E_{out}, E_n$ )
6:    $k = N_{barrier}$  ▷ Calculate inner and outer barrier lines
7:    $[X_{in}, Y_{in}] = \text{GETZHUKLINES}(E_{in}, k)$ 
8:    $[X_{out}, Y_{out}] = \text{GETZHUKLINES}(E_{out}, k)$ 
9:   ▷ Calculate notch line
10:   $k_n = 1$  ▷ Notch has only one barrier line
11:   $[X_n, Y_n] = \text{GETZHUKLINES}(E_n, k_n)$ 
12:  function GETZHUKLINES( $E, k$ )
13:     $C = \sin(p \cdot \theta_E) \cdot \left[ \left( \frac{r_E + \Delta r}{D_{sh}/2} \right)^{2p} - 1 \right] / \left( \frac{r_E + \Delta r}{D_{sh}/2} \right)^p$ 
14:    ▷ Calculate barrier polar angles
15:     $\varphi_{start} = \theta_E$ 
16:     $\varphi_{end} = \tau_{pole}/2$ 
17:    for  $i = 1 : k$  do
18:       $\theta(:, i) = \text{Linspace}(\varphi_{start}(i), \varphi_{end}, N_{points})$ 
19:      ▷ Calculate barrier line vertices
20:       $r = \frac{D_{sh}}{2} \sqrt[p]{\left( C + \sqrt{C^2 + 4 \sin^2(p\theta)} \right) / (2 \sin(p\theta))}$ 
21:       $[X, Y] = \text{pol2cart}(\theta, r)$ 
22:    return  $X, Y$ 
23: Rotate all by  $\alpha_{2Zh} = -\tau_{pole}/2$ 
24: return  $X_{in}, Y_{in}, X_{out}, Y_{out}, X_n, Y_n$ 

```

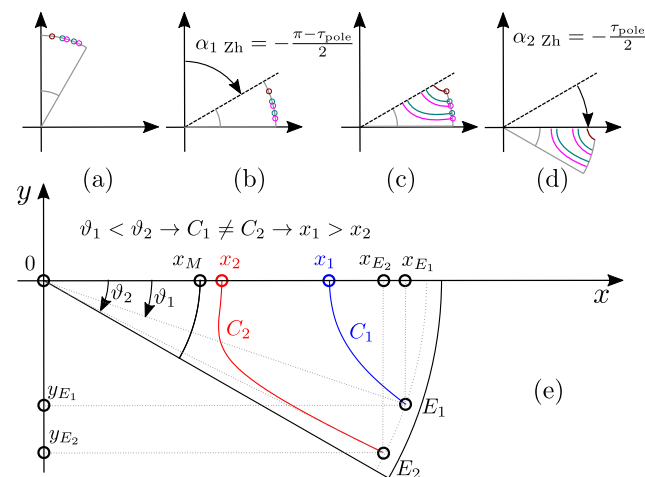


Figure 6. Zhukovsky barrier construction. Subfigures (a)–(e) are explained in the Sections 4.1–4.1.3.

4.1.1. Inner Line Calculation

The first step is the calculation of GETZHUKLINES barrier line function (Algorithm 1: In:7) based on E_{in} and number of barriers k . Note that there are no depth parameters. By definition Zhukovsky barriers are streamlines which cannot mutually intersect. Barrier line profiles (Figure 6e) are derived from conformal mapping theory and the Zhukovsky airflow potential formulation [14,15]. This was originally developed to describe the flow paths of fluids channeled by two infinite plates forming an angle π/p , and a plug centred at the origin of the reference frame. In the solid rotor context, the plug represents the non-magnetic shaft with a radius of $D_{sh}/2$. Equations Algorithm 1: In:12 and Algorithm 1: In:17 express the magnetic field potential lines in parametric form [7].

Next, polar barrier vertices r, θ are calculated based on coefficient vector C , starting point polar coordinates θ_E and number of barriers k (Algorithm 1: In:12–18). Finally, GETZHUKLINES function returns X_{in}, Y_{in} vertices.

4.1.2. Outer Line Calculation

Outer barrier line vertices X_{out}, Y_{out} (Algorithm 1: In:8) are calculated in the same way as inner lines.

4.1.3. Notch Line Calculation

Notch is specific because it has only one barrier line ($k_n = 1$) (Algorithm 1: In:10). The function GETZHUKLINES returns X_n, Y_n which completes the calculation of all barrier lines (Figure 6c). Finally, to be compatible with the rest of the barrier construction procedures, the geometry is rotated by $\alpha_2 Zh = -\tau_{pole}/2$ (Algorithm 1: In:20, Figure 6d).

The next steps (not described within Algorithm 1) are mirroring vertices around horizontal axis (Figure 4e), adding barrier fillets, and geometry rotation around center point (Figure 4f).

Note that Zhukovsky construction pseudo-code is the simplest of all alternatives due to the polar streamline equations (Algorithm 1: In:12, 17).

4.2. Circular Barrier Construction

After definition of all inner and outer barrier line starting points and depth parameters ($E_{in}, E_{out}, E_n, D_{in}, D_{out}, D_n$), Figure 7a, the entire geometry is rotated by $\alpha_1 CrVD = -\pi/2$ (Algorithm 2: In:1, Figure 7b).

Algorithm 2 Construction of circular barriers

```

1: Rotate all by  $\alpha_{1\text{ CrVD}} = -\pi/2$ 
2: Get:  $E_{\text{in}}, E_{\text{out}}, E_n, D_{\text{in}}, D_{\text{out}}, D_n$ 
3:  $[X_{\text{in}}, Y_{\text{in}}, X_{\text{out}}, Y_{\text{out}}, X_n, Y_n] = \text{GETCRLINES}(E_{\text{in}}, E_{\text{out}}, E_n, D_{\text{in}}, D_{\text{out}}, D_n)$ 
4: function GETCRLINES( $E_{\text{in}}, E_{\text{out}}, E_n, D_{\text{in}}, D_{\text{out}}, D_n$ )
5:    $k = N_{\text{barrier}}$  ▷ Calculate inner and outer lines
6:    $[X_{\text{in}}, Y_{\text{in}}, x_{\text{in}}] = \text{GETINNERLINES}(E_{\text{in}}, D_{\text{in}}, k)$ 
7:    $[X_{\text{out}}, Y_{\text{out}}, x_{\text{out}}] = \text{GETOUTERLINES}(E_{\text{out}}, D_{\text{out}}, x_{\text{in}}, k)$ 
   ▷ Calculate notch line
8:    $k_n = 1$  ▷ Notch has only one barrier line
9:    $[X_n, Y_n, x_n] = \text{GETOUTERLINES}(E_n, D_n, x_{\text{out}}(1), k_n)$ 
10:  function GETINNERLINES( $E_{\text{in}}, D_{\text{in}}, k$ )
11:     $x_M = D_{\text{sh}}/2$ 
12:    for  $i = k : 1$  do
13:      if  $i = k$  then
14:        if  $x_{E_{\text{in}}}(i) - x_M < |y_{E_{\text{in}}}(i)|$  then
15:           $x_{\text{in}}(i) = x_M + (x_{E_{\text{in}}}(i) - x_M) \cdot D_{\text{in}}(i)$ 
16:        else
17:           $x_{\text{in}}(i) = x_{E_{\text{in}}}(i) - |y_{E_{\text{in}}}(i)| \cdot D_{\text{in}}(i)$ 
18:      else
19:        if  $x_{E_{\text{in}}}(i) - x_{\text{in}}(i+1) < |y_{E_{\text{in}}}(i)|$  then
20:           $x_{\text{in}}(i) = x_{\text{in}}(i+1) + (x_{E_{\text{in}}}(i) - x_{\text{in}}(i+1)) \cdot D_{\text{in}}(i)$ 
21:        else
22:           $x_{\text{in}}(i) = x_{E_{\text{in}}}(i) - |y_{E_{\text{in}}}(i)| \cdot D_{\text{in}}(i)$ 
23:     $[X_{\text{in}}, Y_{\text{in}}] = \text{GETCRCVTX}(x_{\text{in}}, x_{E_{\text{in}}}, y_{E_{\text{in}}}, k)$ 
24:    return  $X_{\text{in}}, Y_{\text{in}}, x_{\text{in}}$ 
25:  function GETOUTERLINES( $E_{\text{out}}, D_{\text{out}}, x_{\text{in}}, k$ )
26:    for  $i = k : 1$  do
27:      if  $i > 1$  then
28:        if  $x_{E_{\text{out}}}(i) > x_{\text{in}}(i-1)$  then
29:           $x_{\text{out}}(i) = x_{\text{in}}(i) + (x_{\text{in}}(i-1) - x_{\text{in}}(i)) \cdot D_{\text{out}}(i)$ 
30:        else
31:           $x_{\text{out}}(i) = x_{\text{in}}(i) + (x_{E_{\text{out}}}(i) - x_{\text{in}}(i)) \cdot D_{\text{out}}(i)$ 
32:      else
33:        if  $x_{E_{\text{out}}}(i) - x_{\text{in}}(i) < |y_{E_{\text{out}}}(i)|$  then
34:           $x_{\text{out}}(i) = x_{\text{in}}(i) + (x_{E_{\text{out}}}(i) - x_{\text{in}}(i)) \cdot D_{\text{out}}(i)$ 
35:        else
36:           $x_{\text{out}}(i) = x_{E_{\text{out}}}(i) - |y_{E_{\text{out}}}(i)| \cdot D_{\text{out}}(i)$ 
37:     $[X_{\text{out}}, Y_{\text{out}}] = \text{GETCRCVTX}(x_{\text{out}}, x_{E_{\text{out}}}, y_{E_{\text{out}}}, k)$ 
38:    return  $X_{\text{out}}, Y_{\text{out}}, x_{\text{out}}$ 
39:  function GETCRCVTX( $x, x_E, y_E, k$ )
   ▷ Calculate barrier center
40:     $F = (x_E - x_0)^2 + y_E^2 = (x - x_0)^2$ 
41:    for  $i = 1 : k$  do
42:       $x_0 = \text{SOLVE}(F(i))$ 
43:       $R_r = |x - x_0|$ 
   ▷ Calculate barrier polar angles
44:       $\varphi_{\text{start}} = \pi - \text{ATAN2}(y_E, (x_0 - x_E))$ 
45:       $\varphi_{\text{end}} = \pi$ 
46:      for  $i = 1 : k$  do
47:         $\theta(:, i) = \text{Linspace}(\varphi_{\text{start}}(i), \varphi_{\text{end}}, N_{\text{points}})$ 
   ▷ Calculate barrier line vertices
48:       $X = R_r \cdot \cos(\theta) + x_0$ 
49:       $Y = R_r \cdot \sin(\theta)$ 
50:    return  $X, Y$ 
51: return  $X_{\text{in}}, Y_{\text{in}}, X_{\text{out}}, Y_{\text{out}}, X_n, Y_n$ 

```

Algorithm 3 Construction of Hyperbolic barriers

```

1: Rotate all by  $\alpha_{1\text{ HyVE}} = -\pi/2$ 
2: Get:  $E_{\text{in}}, E_{\text{out}}, E_n, D_{\text{in}}, D_{\text{out}}, D_n$ 
3:  $K = 1.1$  ▷ Max eccentricity coefficient
4:  $[X_{\text{in}}, Y_{\text{in}}, X_{\text{out}}, Y_{\text{out}}, X_n, Y_n] = \text{GETHYPLINES}(E_{\text{in}}, E_{\text{out}}, E_n, D_{\text{in}}, D_{\text{out}}, D_n)$ 
5: function GETHYPLINES( $E_{\text{in}}, E_{\text{out}}, E_n, D_{\text{in}}, D_{\text{out}}, D_n$ )
6:    $k = N_{\text{barrier}}$  ▷ Calculate inner and outer barrier lines
7:    $[X_{\text{in}}, Y_{\text{in}}, x_{\text{in}}] = \text{GETINNERLINES}(E_{\text{in}}, D_{\text{in}}, k)$ 
8:    $[X_{\text{out}}, Y_{\text{out}}, x_{\text{out}}] = \text{GETOUTERLINES}(E_{\text{out}}, D_{\text{out}}, x_{\text{in}}, k)$ 
   ▷ Calculate notch line
9:    $k_n = 1$  ▷ Notch has only one barrier line
10:   $[X_n, Y_n, x_n] = \text{GETOUTERLINES}(E_n, D_n, x_{\text{out}}(1), k_n)$ 
11:  function GETINNERLINES( $E_{\text{in}}, D_{\text{in}}, k$ )
12:     $x_M = D_{\text{sh}}/2$ 
13:    for  $i = k : 1$  do
14:      if  $i = k$  then
15:         $e_{\text{min}}(i) = (r_{E_{\text{in}}}(i) - x_M) / (x_{E_{\text{in}}}(i) - x_M)$ 
16:      else
17:         $e_{\text{min}}(i) = \frac{r_{E_{\text{in}}}(i) - x_{\text{in}}(i+1)}{x_{E_{\text{in}}}(i) - x_{\text{in}}(i+1)}$ 
18:         $e_{\text{max}}(i) = K \cdot e_{\text{min}}(i)$  ▷ Est. max eccentricity
19:         $e_{\text{in}}(i) = (e_{\text{max}}(i) - e_{\text{min}}(i)) \cdot D_{\text{in}}(i) + e_{\text{min}}(i)$ 
20:         $x_{d_{\text{in}}}(i) = \frac{r_{E_{\text{in}}}(i)}{e_{\text{in}}(i)} - x_{E_{\text{in}}}(i)$  ▷ Left directrix
21:         $x_{\text{in}}(i) = e_{\text{in}}(i) \cdot x_{d_{\text{in}}}(i) / (1 + e_{\text{in}}(i) \cdot \text{sgn}(x_{d_{\text{in}}}(i)))$ 
22:       $[X_{\text{in}}, Y_{\text{in}}] = \text{GETHYPVTX}(x_{d_{\text{in}}}, e_{\text{in}}, \theta_{E_{\text{in}}}, k)$ 
23:  return  $X_{\text{in}}, Y_{\text{in}}, x_{\text{in}}$ 
24:  function GETOUTERLINES( $E_{\text{out}}, D_{\text{out}}, x_{\text{in}}, k$ )
25:    for  $i = k : 1$  do
26:       $e_{\text{min}}(i) = \frac{r_{E_{\text{out}}}(i) - x_{\text{in}}(i)}{x_{E_{\text{out}}}(i) - x_{\text{in}}(i)}$ 
27:      if  $i > 1$  then
28:         $e_{\text{max}}(i) = \frac{r_{E_{\text{out}}}(i) - x_{\text{in}}(i+1)}{x_{E_{\text{out}}}(i) - x_{\text{in}}(i+1)}$ 
29:      else
30:         $e_{\text{max}}(i) = K \cdot e_{\text{min}}(i)$  ▷ Est. max eccentricity
31:         $e_{\text{out}}(i) = (e_{\text{max}}(i) - e_{\text{min}}(i)) \cdot D_{\text{out}}(i) + e_{\text{min}}(i)$ 
32:         $x_{d_{\text{out}}}(i) = \frac{r_{E_{\text{out}}}(i)}{e_{\text{out}}(i)} - x_{E_{\text{out}}}(i)$  ▷ Left directrix
33:         $x_{\text{out}}(i) = \frac{e_{\text{out}} \cdot x_{d_{\text{out}}}(i)}{1 + e_{\text{out}}(i) \cdot \text{sgn}(x_{d_{\text{out}}}(i))}$ 
34:       $[X_{\text{out}}, Y_{\text{out}}] = \text{GETHYPVTX}(x_{d_{\text{out}}}, e_{\text{out}}, \theta_{E_{\text{out}}}, k)$ 
35:  return  $X_{\text{out}}, Y_{\text{out}}, x_{\text{out}}$ 
36:  function GETHYPVTX( $x_d, e, \theta_E, k$ )
   ▷ Calculate barrier polar angles
37:     $\varphi_{\text{start}} = 0; \varphi_{\text{end}} = \theta_E$ 
38:    for  $i = 1 : k$  do
39:       $\theta(:, i) = \text{Linspace}(\varphi_{\text{start}}, \varphi_{\text{end}}(i), N_{\text{points}})$ 
   ▷ Calculate barrier line vertices
40:     $r = e \cdot x_d / (1 + e \cdot \text{sgn}(x_d) \cdot \cos(\theta))$ 
41:     $[X, Y] = \text{pol2cart}(\theta, r)$ 
42:  return  $X, Y$ 
43: return  $X_{\text{in}}, Y_{\text{in}}, X_{\text{out}}, Y_{\text{out}}, X_n, Y_n$ 

```

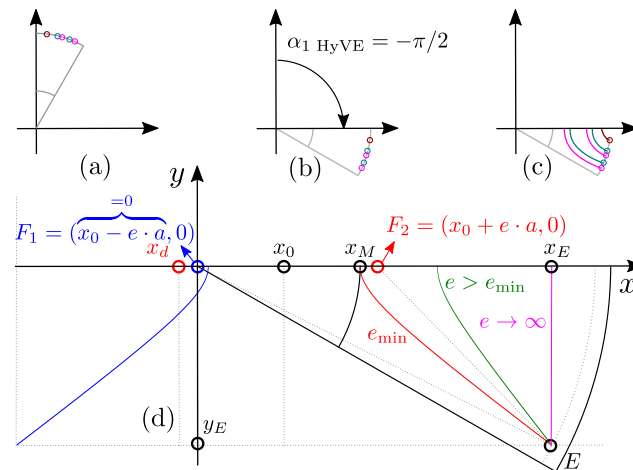


Figure 8. Hyperbolic barrier construction. Subfigures (a)–(d) are explained in the Sections 4.3–4.3.3.

4.3.1. Inner Line Calculation

Lets first consider GETINNERLINES function which calculates inner lines vertices (X_{in} , Y_{in}) and intersections (x_{in}) based on E_{in} , D_{in} and number of barriers k (Algorithm 3: Ln:7).

To simplify the construction, the left focus is constructed in the origin of the coordinate system (Figure 8d, blue). The intersections with $y = 0$ (Figure 4d) are critical for barrier calculation and they depend on the minimum eccentricity Equation $e_{min} = (r_E - x) / (x_E - x)$, x is the inner or outer intersection limit.

In case of the most inner line ($i = k$), the feasibility limit is rotor shaft (x_M), the minimum eccentricity $e_{min}(i)$ is calculated via Algorithm 3: Ln:15. For the second most inner line ($i = k - 1$), $e_{min}(i + 1)$ is calculated via Algorithm 3: Ln:17.

If particular barrier has eccentricity within the limits $e \in [e_{min}, \infty]$, the construction is feasible (Figure 8d, green). Otherwise the barrier is infeasible (Figure 8d, red, magenta). This is why it is important to limit maximum eccentricity (Algorithm 3: Ln:18), where $K = 1.1$ is empirically determined coefficient. The final eccentricity is then calculated depending of the depth parameter (Algorithm 3: Ln:19). The next steps are calculation of the left directrix (Algorithm 3: Ln:20) and intersection point (x_{in}) based on polar hyperbola Equation (Algorithm 3: Ln:21).

All inner line intersections are then iteratively calculated following the described procedure (Algorithm 3: Ln:13–21).

Next, GETHYPTX function (Algorithm 3: Ln:22) based on calculated directrices $x_{d_{in}}$, eccentricities e_{in} , angular starting point coordinates $\theta_{E_{in}}$ and k , calculates the hyperbolic vertices in polar coordinates r , θ , and returns barrier vertices X_{in} , Y_{in} (Algorithm 3: Ln:36–41).

Finally, GETINNERLINES returns X_{in} , Y_{in} , x_{in} .

4.3.2. Outer line calculation

At this point all inner barrier intersections x_{in} are defined and now present feasibility limits for outer barrier line construction (Algorithm 3: Ln:8). The rest of the GETOUTERLINES code (Algorithm 3: Ln:24–34) is executed in the similar manner as in GETINNERLINES. The function returns X_{out} , Y_{out} , x_{out} .

4.3.3. Notch line calculation

Notch is specific because it has only one barrier line ($k_n = 1$) and depends on the most outer intersection $x_{out}(1)$ (Algorithm 3: Ln:10). Finally, GETOUTERLINES returns X_n , Y_n , x_n which completes the calculation of all barrier lines (Figure 7c).

The next steps (not described within Algorithm 3) are mirroring vertices around horizontal axis (Figure 4e), adding barrier fillets, and geometry rotation around center point (Figure 4f).

5. Conformal Modifications

5.1. Conformal Mapping

A conformal or angle-preserving transformation also called conformal mapping is a transformation $w = f(z)$ that preserves local angles. An analytic function is conformal at any point where it has a nonzero derivative [17].

Conformal transformations can prove extremely useful in solving physical problems. If the selected complex function $w = f(z)$ satisfies the condition that the real and imaginary parts of w satisfy the Cauchy–Riemann equations and Laplace’s equation, they automatically provide a scalar potential and a so-called stream function [17] (e.g., Zhukovski barrier lines, Figure 1d, red).

FEA is typically used for electromagnetic performance calculations in electric machine design. Depending on the complexity of the calculation and mesh density, it can take several minutes before the calculation is completed. The simulation time can be reduced if a conformal mapping is used for calculation of the analytical airgap magnetic field [16]. The method conformally transforms electric machine cross-section to w -plane thus enabling quick analytic calculations. The w -plane results are then inversely mapped to the real plane. Compared to FEA simulation, this approach yields results within seconds [11,12,16]. On the other hand, the disadvantage is the difficult implementation on complex rotor geometries.

This paper proposes a method for geometrical modification of any SyRM barrier geometry using conformal mapping.

5.2. Mapping Workflow

Generated rotor barrier lines are defined by sorted vertices $(X_{in}, Y_{in}, X_{out}, Y_{out}, X_n, Y_n)$ containing corresponding x, y coordinates which can be drawn on a 2D real Euclidean plane (Figure 9a). For easier manipulation purposes, real plain coordinates are redefined in complex z -plane ($z = x + jy$), Figure 9b. Considering that the vertices are the same in real and complex plane, this is a trivial transformation.

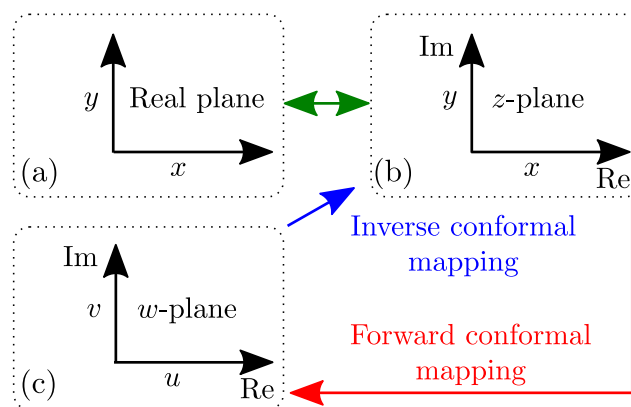


Figure 9. Mapping workflow, Subfigures (a)–(c) are explained within the Section 5.2.

SyRM rotor has a circular layout which can be exploited by selecting the convenient complex function $f(z)$ and applying forward conformal mapping to a complex w -plane ($w = u + jv$), Figure 9c. Geometrical modifications of the barrier geometry are then performed in w -plane and upon completion, returned back to z (and real) plane via inverse conformal mapping. The benefit of the approach is easier barrier modification which leads to simplified software coding.

5.3. Complex Functions

The principal objects of study are complex-valued functions $f(z)$, depending on a single complex variable $z = x + jy \in \mathbb{C}$. In general, the function $f : \Omega \rightarrow \mathbb{C}$ is defined on an open subdomain, $z \in \Omega \subset \mathbb{C}$, of the complex plane. Any complex function can be uniquely written as a complex combination $f(z) = f(x + iy) = \text{Re } u(x, y) + j\text{Im } v(x, y)$ [18].

5.3.1. Forward Conformal Mapping

When working with circular shapes, it is useful to use polar form $z = re^{j\theta}$. Considering that electric machines can have an arbitrary number of pole pairs, it is very convenient to select a complex function that somehow ignores angle changes when the number of poles is increased or decreased. This kind of functionality can be achieved by a complex natural logarithm. In terms of polar coordinates, the complex logarithm has the form $w = \ln z = \ln(re^{j\theta}) = \ln r + \ln e^{j\theta} = \ln r + j\theta$.

Thus, the logarithm of a complex number has a real part which is a well-defined harmonic function save for a logarithmic singularity at the origin $x = y = 0$. The imaginary part of the complex logarithm is the polar angle, known in complex analysis as the *phase*.

$$u(x, y) = \operatorname{Re}(\ln z) = \ln r = \frac{1}{2} \ln(x^2 + y^2) \quad (2)$$

$$v(x, y) = \operatorname{Im}(\ln z) = \theta = \arctan \frac{y}{x} \quad (3)$$

Due to inherent symmetry, it is enough to analyze one electric machine pole. A minimum number of poles is 2, which equals to $[0, \pi]$ radian angular span in z -plane, which is mapped to the same vertical span in w -plane. Figure 10 illustrates mapping of different combination of pole geometries. To summarize, $w = \ln z$ always maps z to the upper half-plane with vertical boundaries $v \in [0, \pi/p]$ (p is the number of pole pairs) and horizontal boundaries of $u \in [\ln x_M, \ln R]$.

Note that $\ln z$ always maps SyRM barrier lines horizontally (Figure 10, w -plane), regardless of the number of poles. This feature is exploited for the creation of the Modified Zhukovsky variable depth barrier lines (MZhVD).

5.3.2. Inverse Conformal Mapping

Inverse function for return to z -plane is a complex exponential $z = e^w = e^{u+jv}$. Since $w \in \mathbb{C}$ is a non-zero complex number, the equation can be written as $z = e^u \cdot (\cos vs. + j \sin v)$ with real and imaginary parts equal to:

$$x(u, v) = \operatorname{Re}(e^w) = e^u \cdot \cos v \quad (4)$$

$$y(u, v) = \operatorname{Im}(e^w) = e^u \cdot \sin v \quad (5)$$

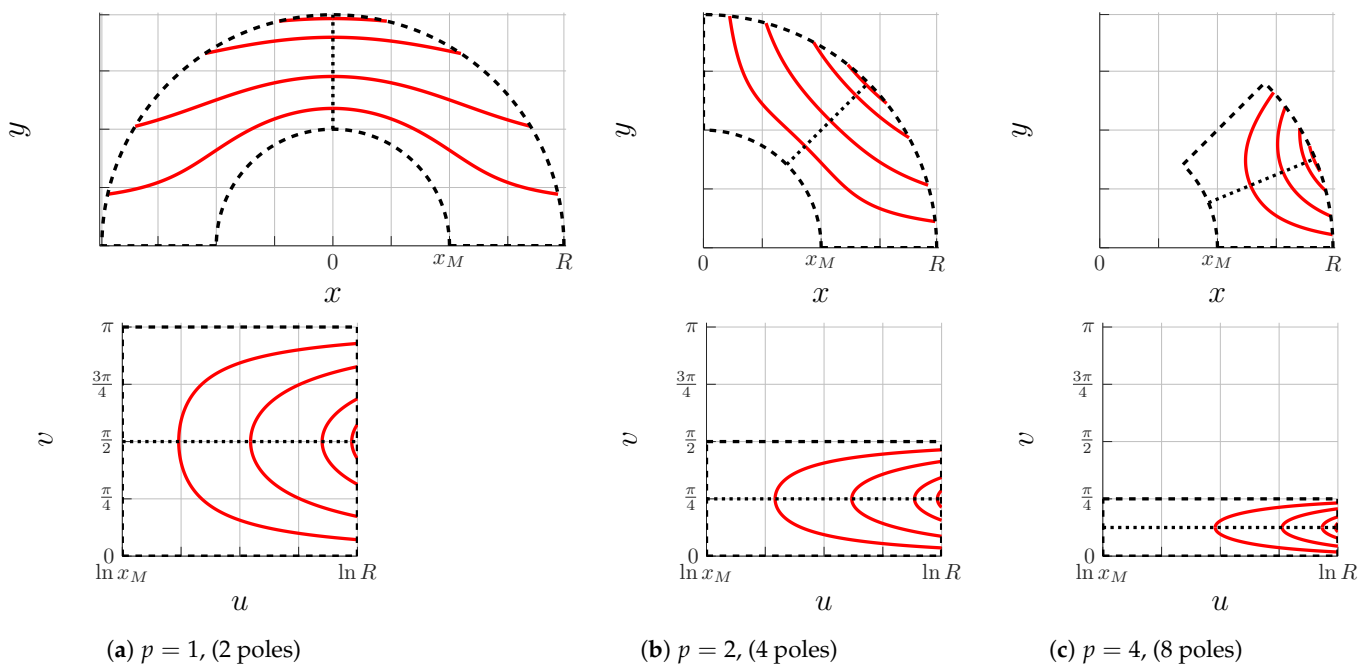


Figure 10. Real z -plane (upper row); Conformal mapping to $w = \ln z$ plane (bottom row).

5.4. Depth Modification

To summarize, original Zhukovsky barrier lines ($X_{in}, Y_{in}, X_{out}, Y_{out}, X_n, Y_n$) with corresponding x, y vertices are written as $z = x + jy$ (Figure 11a, red) and conformally transformed to w -plane via $w = \ln z$ as $w = u + jv$ (Figure 11b, red). As previously mentioned, Zhukovsky lines cannot mutually intersect. To secure barrier depth variability and improve machine performance, we are introducing Zhukovsky barrier depth modification (MZhVD) via dimensionless depth parameters D_{in}, D_{out}, D_n .

Depth variability is secured via addition of cosine offsets to the w -plane barrier lines according to Equations (6) and (7). In theory, any even function can be used for generation of Δ offset (7), cosine has been selected due to implementation simplicity.

$$T = 2 \cdot [\text{Max}(V_{Zhk}) - \text{Min}(V_{Zhk})] \quad (6)$$

$$\Delta = \Delta_{Dpth} \cdot \cos \left[\frac{\pi}{T} (V_{Zhk} - v_E) - \pi/2 \right] \quad (7)$$

Upon modification (Figure 11b, blue), barrier lines are inversely mapped to z -plane via function $z = e^w$ (Figure 11a, blue). The main benefit of the proposed procedure is a simplified modification of SyRM barriers without any influence on simulation time. The following section explains the modification procedure step by step.

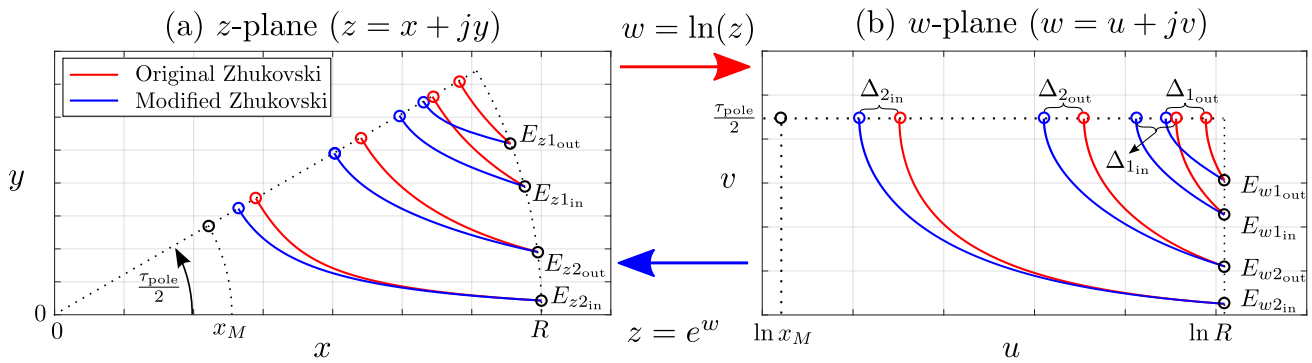


Figure 11. Modification of Zhukovski lines via conformal mapping. Subfigures (a) and (b) are explained within the Section 5.4 and related subsections.

5.5. Modified Zh Barrier Construction

After definition of all inner and outer barrier line starting points (E_{in}, E_{out}, E_n), Figure 12a, the entire geometry is rotated by $\alpha_{1\text{MZhVD}} = -(\pi/2 - \tau_{\text{pole}}/2)$ (Algorithm 4: Ln:2, Figure 12b).

5.5.1. Inner Line Calculation

The first step is the calculation of GETINNERLINES barrier line function (Algorithm 4: Ln:9). Inner line vertices (X_{in}, Y_{in}) are calculated based on E_{in}, D_{in} and number of barriers k .

Next (Algorithm 4: Ln:14), original Zhukovsky barrier (Zh) vertices ($X_{Zh_{kin}}, Y_{Zh_{kin}}$) are calculated in the same way as in Algorithm 1, Figure 12c. Original vertices ($X_{Zh_{kin}}, Y_{Zh_{kin}}$) and barrier starting points ($x_{E_{in}}, y_{E_{in}}$) are then conformally mapped to w -plane as $u_{Zh_{kin}}, v_{Zh_{kin}}$ and $u_{E_{in}}, v_{E_{in}}$, Algorithm 4: Ln:15, 16, Figure 12d, dotted lines.

Intersections u_{in} are critical for Modified Zhukovsky barrier variable depth (MZhVD) calculation and they are fully tied to original Zhukovsky intersections $u_{Zh_{kin}}$ (calculated via Algorithm 4: Ln:17). As previously mentioned, original Zhukovsky barriers cannot mutually intersect which makes them absolutely feasible, but on the other hand, barrier depth variability cannot be achieved.

Modified Zhukovsky depth variation is secured by iterative recalculation of intersection points in w -plane. In case of the most inner line ($i = k$), the barrier depth limit is rotor shaft (u_M), and the original Zhukovsky intersection $u_{Zh_{kin}}(i)$ (Algorithm 4: Ln:20). For the second most inner line ($i = k - 1$), depth limit is $u_{Zh_{kin}}(i + 1)$ and the intersection is

calculated via (Algorithm 4: ln:22). All inner line intersections are then iteratively calculated (Algorithm 4: ln:18–22).

Next, inner line depth offset is calculated based on original Zhukovsky and calculated inner intersections (Algorithm 4: ln:23). CLCDPTH function (Algorithm 4: ln:25) calculates modified vertices in w -plain (Figure 12d, full lines), performs inverse conformal mapping, and returns z -plain vertices (Algorithm 4: ln:59–64, Figure 12e).

Finally, GETINNERLINES function returns X_{in}, Y_{in} vertices.

5.5.2. Outer Line Calculation

At this point all w -plane inner barrier intersections u_{in} are defined and now present limits for outer barrier line construction. The rest of the GETOUTERLINES code (Algorithm 4: ln:27–40) is executed in the similar manner as in GETINNERLINES. The function at the end returns $X_{out}, Y_{out}, x_{out}$.

5.5.3. Notch Line Calculation

Notch is specific because it has only one barrier line ($k_n = 1$) and depends on the most outer w -plain intersection $u_{out}(1)$ (Algorithm 4: ln:12). Finally, GETOUTERLINES returns X_n, Y_n, u_n which completes the calculation of all barrier lines (Figure 12e). Finally, to be compatible with the rest of the barrier construction procedures, the geometry is rotated by $\alpha_2 MZhVD = -\tau_{pole}/2$ (Algorithm 4: ln:65, Figure 12f).

The next steps (not described within Algorithm 4) are mirroring vertices around horizontal axis (Figure 4e), adding barrier fillets, and geometry rotation around center point (Figure 4f).

Algorithm 4 Construction of Modified Zhukovsky barriers

```

1:  $\tau_{pole} = 2\pi/(2p)$ 
2: Rotate all by  $\alpha_1 MZhVD = -(\pi/2 - \tau_{pole}/2)$ 
3: Get:  $E_{in}, E_{out}, E_n$ 
4:  $\Delta r = 0$ 
5:  $u_M = \ln(x_M)$  ▷ Shaft limit in  $w$ -plane
6:  $[X_{in}, Y_{in}, X_{out}, Y_{out}, X_n, Y_n] = \text{GETHYPLINES}(E_{in}, E_{out}, E_n, D_{in}, D_{out}, D_n)$ 
7: function GETMODZHLINES( $E_{in}, E_{out}, E_n, D_{in}, D_{out}, D_n$ )
8:    $k = N_{barrier}$  ▷ Calculate inner and outer barrier lines
9:    $[X_{in}, Y_{in}, u_{in}] = \text{GETINNERLINES}(E_{in}, D_{in}, k)$ 
10:   $[X_{out}, Y_{out}, u_{out}] = \text{GETOUTERLINES}(E_{out}, D_{out}, u_{in}, k)$ 
11:  ▷ Calculate notch line
12:   $k_n = 1$  ▷ Notch has only one barrier line
13:   $[X_n, Y_n] = \text{GETOUTERLINES}(E_n, D_n, u_{out}(1), k_n)$ 
14:  function GETINNERLINES( $E_{in}, D_{in}, k$ )
15:     $[X_{Zhkin}, Y_{Zhkin}] = \text{GETZHUKLINES}(E_{in}, k)$ 
16:    ▷ Forward conformal transformation
17:     $[U_{Zhkin}, V_{Zhkin}] = \text{FRWCONF}(X_{Zhkin}, Y_{Zhkin})$ 
18:     $[u_{E_{in}}, v_{E_{in}}] = \text{FRWCONF}(x_{E_{in}}, y_{E_{in}})$ 
19:    ▷ Original Zhukovsky inner line intersection limits
20:     $u_{Zhkin} = U_{Zhkin}(\text{find}(\max(V_{Zhkin})))$ 
21:    for  $i = k : 1$  do
22:      if  $i = k$  then
23:         $u_{in}(i) = u_M + (u_{Zhkin}(i) - u_M) \cdot D_{in}(i)$ 
24:      else
25:         $u_{in}(i) = u_{Zhkin}(i + 1) + (u_{Zhkin}(i) - u_{Zhkin}(i + 1)) \cdot D_{in}(i)$ 
26:     $\Delta_{Dpthin} = u_{Zhkin} - u_{in}$ 
27:     $[X_{in}, Y_{in}] =$ 
28:      CLCDPTH( $\Delta_{Dpthin}, U_{Zhkin}, V_{Zhkin}, v_{E_{in}}$ )
29:  return  $X_{in}, Y_{in}, u_{in}$ 

```

Algorithm 4 Cont.

```

27: function GETOUTERLINES( $E_{out}, D_{out}, u_{in}, k$ )
28:   [ $X_{Zhkout}, Y_{Zhkout}$ ] = GETZHUKLINES( $E_{out}, k$ )
   ▷ Forward conformal transformation
29:   [ $U_{Zhkout}, V_{Zhkout}$ ] = FRWCONF( $X_{Zhkout}, Y_{Zhkout}$ )
30:   [ $u_{Eout}, v_{Eout}$ ] = FRWCONF( $x_{Eout}, y_{Eout}$ )
   ▷ Original Zhukovsky outer line intersection limits
31:    $u_{Zhkout} = U_{Zhkout}(\text{find}(\max(V_{Zhkout})))$ 
32:   for  $i = k : 1$  do
33:     if  $i > 1$  then
34:        $u_{out}(i) = u_{in}(i) + (u_{in}(i-1) - u_{in}(i)) \cdot D_{out}(i)$ 
35:     else
36:        $u_{out}(i) = u_{in}(i) + (u_{Zhkout}(i) - u_{in}(i)) \cdot D_{out}(i)$ 
37:    $\Delta_{Dpth_{out}} = u_{Zhkout} - u_{out}$ 
38:   [ $X_{out}, Y_{out}$ ] =
39:   CLCDPTH( $\Delta_{Dpth_{out}}, U_{Zhkout}, V_{Zhkout}, v_{Eout}$ )
40: return  $X_{out}, Y_{out}, u_{out}$ 
41: function GETZHUKLINES( $E, k$ )
42:    $C = \sin(p \cdot \vartheta_E) \cdot \left[ \left( \frac{r_E + \Delta r}{D_{sh}/2} \right)^{2p} - 1 \right] / \left( \frac{r_E + \Delta r}{D_{sh}/2} \right)^p$ 
   ▷ Calculate barrier polar angles
43:    $\varphi_{start} = \vartheta_E$ 
44:    $\varphi_{end} = \tau_{pole}/2$ 
45:   for  $i = 1 : k$  do
46:      $\vartheta(:, i) = \text{Linspace}(\varphi_{start}(i), \varphi_{end}, N_{points})$ 
   ▷ Calculate barrier line vertices
47:    $r = \frac{D_{sh}}{2} \sqrt[p]{\left( C + \sqrt{C^2 + 4 \sin^2(p\vartheta)} \right) / (2 \sin(p\vartheta))}$ 
48:   [ $X, Y$ ] = pol2cart( $\vartheta, r$ )
49: return  $X, Y$ 
50: function FRWCONF( $X, Y$ )
51:   [ $r, \theta$ ] = cart2pol( $X, Y$ )
52:    $U = \ln r$ 
53:    $V = \theta$ 
54: return  $U, V$ 
55: function INVCONF( $U, V$ )
56:    $X = e^U \cdot \cos V$ 
57:    $Y = e^U \cdot \sin V$ 
58: return  $X, Y$ 
59: function CLCDPTH( $\Delta_{Dpth}, U_{Zhk}, V_{Zhk}, v_E$ )
60:    $T = 2 \cdot (\text{Max}(V_{Zhk}) - \text{Min}(V_{Zhk})); f = \frac{1}{2T}$ 
61:    $\Delta = \Delta_{Dpth} \cdot \cos(2\pi f(V_{Zhk} - v_E) - \pi/2)$ 
62:    $U_{new} = U_{Zhk} - \Delta$ 
   ▷ Inverse conformal transformation
63:   [ $X, Y$ ] = INVCONF( $U_{new}, V_{Zhk}$ )
64: return  $X, Y$ 
65: Rotate all by  $\alpha_2 \text{ MZhvD} = -\tau_{pole}/2$ 
66: return  $X_{in}, Y_{in}, X_{out}, Y_{out}, X_n, Y_n$ 

```

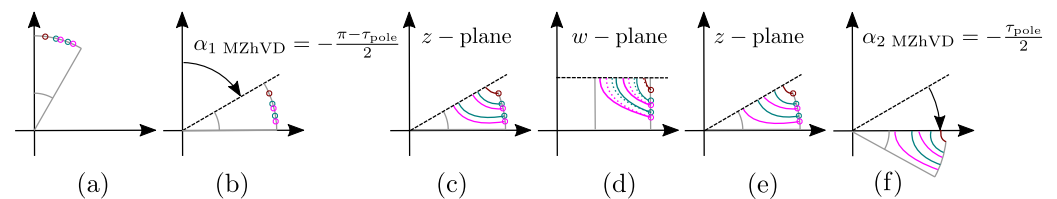


Figure 12. Modified Zhukovsky barrier construction and rotation steps. Subfigures (a)–(f) are explained within the Sections 5.5–5.5.3.

6. Parametric Complexity

A high number of optimization variables is associated with a longer optimization time [19,20], so the imperative is to simplify barrier topology parametrization. Gamba et al. [14] state that three parameters per barrier (total complexity of $3k$, where k is the number of barriers) are the appropriate number for a fast yet accurate description of multi-barrier SyRM (barrier fillet parameters are not included). In the previous publication, our group of authors has reduced the complexity to $2 \cdot k + 1$ per barrier [7].

Table 3 shows the calculation of total number of SyRM parameters for the demonstrated pseudo-code. Note that the parameters $\vartheta_{1_{out}}, \vartheta_{k_{in}}$ are subtracted from the count because they are constant and equal to zero (Table 1).

Table 4 lists the complexity comparison of the presented procedures and similar approaches in [7,10,14]. The examples in [7,10,14] do not have a notch feature, so to have a fair comparison, the notch is not included in the complexity calculation (Table 3). Compared to [14], and [7], Zh, respectively, yields smaller complexity ($2k$), while CrVD, HyVE, MZhVD have the same complexity as in [10].

Overall, the construction principle explained in Section 3 enables the higher degree of design freedom. Considering that the simple barrier topologies are sub-optimal compared to more complex types, developing the set of different parametrization methods with equal parametric complexity is certainly a novel contribution.

Table 3. Calculation of total parameter number. Color coding is according to Figure 4 variables.

Sum:	Description	Symbol	Topology
1	Min. angle	ϑ_{min}	
2	Max. angle	ϑ_{max}	
$k + 2$	Barrier angle in	$\vartheta_{1..k_{in}}$	
$2k + 2$	Barrier angle out	$\vartheta_{1..k_{out}}$	
$2k + 2 - 2$	Remove constants	$\vartheta_{1_{out}} = \vartheta_{k_{in}} = 0$	
$2k$	-	-	Zh
$4k$	Barrier depths	$D_{1..k_{in}}$	HyVE
	Barrier depths	$D_{1..k_{out}}$	CrVD
			MZhVD

Table 4. Parametric complexity.

Topology	Complexity	$k = 2$	$k = 3$	$k = 4$
Zhukovsky; Gamba et al. [14]	$3k$	6	9	12
Circular; Stipetic et al. [10]	$4k$	8	12	16
Zhukovsky; Ban et al. [7]	$2k + 1$	5	7	9
Zh	$2k$	4	6	8
HyVE				
CrVD	$4k$	8	12	16
MZhVD				

7. Pseudo-Code Validation

After the implementation of the proposed pseudo-code, a set of questions naturally arises. Which barrier topology yields the best performance for the given requirements? Is Modified Zhukovsky barrier type better than alternative topologies? For this reason, we have conducted a detailed optimization study based on meta-modeling (surrogate modeling) approach which compared the different barrier topologies (details are available in [21]).

The optimization process couples automated geometry generation (Matlab), electromagnetic finite element analysis (Ansys Motor-CAD), and metamodel optimization (Ansys OptiSlang). Seven rotor topologies have been derived from circular, hyperbolic, and Zhukovsky barrier types:

1. Circular concentric (CrC)
2. Circular variable depth (CrVD)
3. Hyperbolic with fixed eccentricity (HyFE)
4. Hyperbolic with variable eccentricity (HyVE)
5. Original Zhukovsky (Zh)
6. Modified Zhukovsky variable depth (MZhVD)
7. Modified Zhukovsky with equal barrier depth (MZhED)

The same optimization strategy (maximize torque per volume (TPV), minimize losses) has been applied to all variants, and results prove that barrier type substantially affects the final machine performance. For easier comparison, seven designs (one per topology) with approximately the same losses (5200 W) have been selected (Figure 13, Table 5).

Performance wise, HyFE topology yields the worst results and is considered as baseline design (Gain = 0%). Performance gain is calculated via: $\text{Gain} = (T_{\text{avg}}/T_{\text{HyFE avg}} - 1) \cdot 100\%$. The best results are achieved by MZhVD topology. In relation to the worst (baseline) topology, the performance gain is 14.9% and the power factor is increased from 0.61 to 0.69. It is important to note that these comparisons are valid for design requirements presented in [21]. Other combinations of optimization objectives and requirements might yield a different results.

HyFE, CrC and MZhED are special case topologies already covered in CrVD, HyVE and MzVD pseudo-code. Considering that stated topologies can be achieved by appropriate combination of barrier depth parameters, they are not considered in this paper. The summary of three best optimized cross-sections are provided on Figure 14.

Table 5. Optimization result comparison table [21].

Name	Unit	HyFE	CrC	HyVE	CrVD	Zh	MZhED	MZhVD
TPV	Nm/dm ³	32.5	33.1	34.3	35.4	36.2	36.4	37.3
V_{active}	dm ³	6.47	6.47	6.47	6.47	6.47	6.47	6.47
P_{loss}	kW	5188	5199	5209	5182	5188	5197	5184
P_{mech}	kW	37.4	38.1	39.5	40.8	41.7	41.9	43.0
T_{avg}	Nm	210.1	214.2	221.9	229.0	234.1	235.6	241.3
$T_{\text{ripp.}}$	%	12.1	14.1	11.7	12.7	9.7	9.3	13.7
n	rpm	1700	1700	1700	1700	1700	1700	1700
l_s	mm	180	180	180	180	180	180	180
γ	°	57.9	60.3	61.4	62.5	61.8	61.8	62.9
I_{max}	A _{rms}	95.6	95.6	94.3	94.1	95.9	95.7	95.7
$\cos \varphi$	-	0.61	0.62	0.66	0.67	0.67	0.67	0.69
Gain	%	0.0	1.9	5.6	9.0	11.4	12.1	14.9

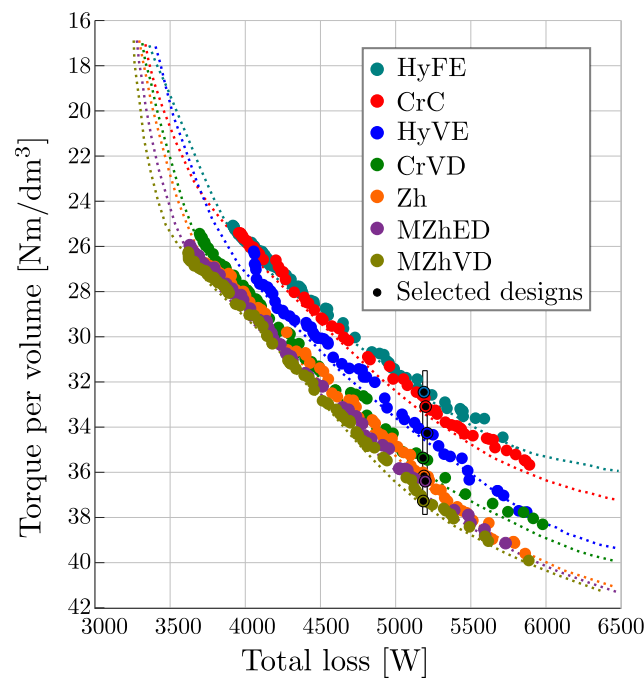


Figure 13. Validated Pareto fronts for each design variant [21].

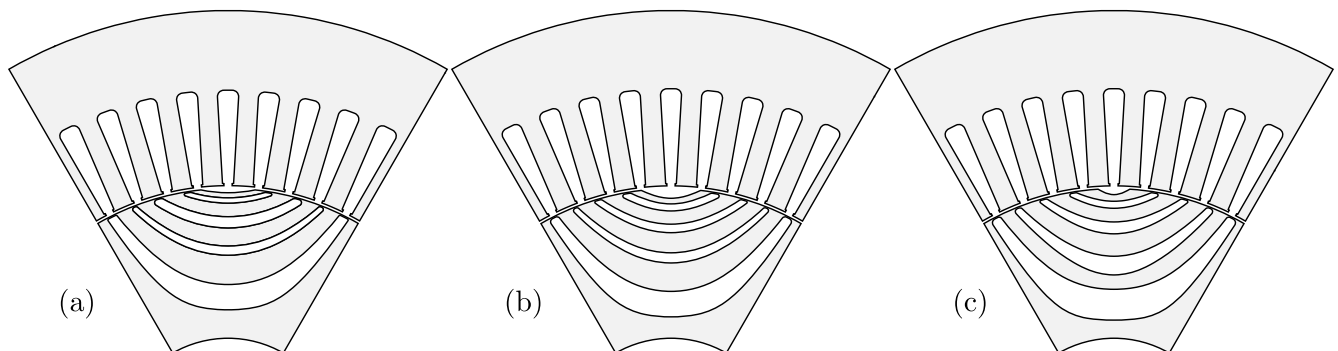


Figure 14. Optimized cross section of three best topologies: (a) Zh; (b) MZhED; (c) MZhVD [21].

8. Conclusions

SyRM barrier generation procedure was studied in detail. A pseudo-code solution that secures absolute feasibility, barrier topology complexity minimization, and simple implementation is provided. Four smooth barrier types have been presented: circular variable depth (CrVD), hyperbolic with variable eccentricity (HyVE), original Zhukovsky (Zh) and modified Zhukovsky with variable depth (MZhVD). Absolute feasibility is a very important feature because it enables the use of dimensionless parameters which secure code robustness and design scalability to any physical dimension. Barrier topology complexity has been minimized via a systematic approach to design automation (Section 3) and careful analysis of construction features of each topology.

HyVE and CrVD topologies are more complex $4k$ while Zh has smaller complexity $2k$. On the other hand, in its original form, Zh type does not support barrier depth variability which can be a design drawback. Therefore, we introduce a novel MZhVD topology based on geometrical conformal mapping of the original Zh design. This modification provides greater design freedom and sets MZhVD complexity to $4k$ (same as HyVE and CrVD).

Compared with other referenced topologies, the presented solutions offer higher design freedom with smaller or equal parametric complexity.

It should be noted that construction of barrier fillets (Figure 4f) has not been covered in this paper. Adding a precise fillet between two discrete intersecting lines is a rather

complex problem, which deserves a stand-alone publication. A minimum example code with detailed instructions is planned to be published in the near future.

Overall, the presented pseudo-code provides a valuable starting point for the designer who wants to investigate different SyRM smooth barrier topologies.

The description and list of used variables has been added in the Appendix A.

Author Contributions: Conceptualization, B.B.; methodology, B.B.; validation, B.B. and S.S.; original draft preparation, B.B.; review and editing, S.S.; visualization, B.B.; supervision, S.S.; funding acquisition, S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Croatian Science Foundation under the project IP-2018-01-5822-HYDREL.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This paper is an extension of Branko Ban's Ph.D. research work on Synchronous Reluctance machines mentored by Stjepan Stipetic.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Abbreviation	Description
EV	Electric vehicle
FEA	Finite element analysis
IPM	Interior permanent magnet
IM	Induction machine
CrC	Circular concentric barrier
CrVD	Circular variable depth barrier
HyFE	Hyperbolic fixed eccentricity barrier
HyVE	Hyperbolic variable eccentricity barrier
Zh	Original Zhukovsky barrier
MZhed	Modified Zhukovsky equal depth barrier
MZhVD	Modified Zhukovsky variable depth barrier
PM	Permanent magnet
PTO	Power take off
e-PTO	Electric power take off
SyRM	Synchronous reluctance machine
TPV	Torque per volume

Appendix A. Variable List

All variables have been listed by the order of appearance within text. Vector variables are bolded. e.g., \mathbf{R} is a variable vector, while R represents a scalar variable.

Table A1. List of variables.

No.	Variable	Description	No.	Variable	Description
1	$w_{c \min}$	Minimum flux carrier width	33	$\alpha_{1 \text{ Zh}}$	First rotation angle in Zh generation
2	D_{in}	Inner barrier depth parameters	34	$\alpha_{2 \text{ Zh}}$	Second rotation angle in Zh generation
3	D_{out}	Outer barrier depth parameters	35	$\alpha_{1 \text{ CrVD}}$	First rotation angle in Zh generation
4	D_n	Notch depth parameter	36	$\alpha_{1 \text{ HyVE}}$	First rotation angle in Zh generation
5	p	Number of pole pairs	37	$\alpha_{1 \text{ MZhVD}}$	First rotation angle in MZhVD generation
6	k	Number of flux barriers	38	$\alpha_{2 \text{ MZhVD}}$	Second rotation angle in MZhVD generation
7	ϑ_{\min}	Minimum angular barrier span	39	\mathbf{r}	Radial Zhukovsky line coordinate vector
8	ϑ_{\max}	Maximum angular barrier span	40	$\boldsymbol{\theta}$	Angular Zhukovsky line coordinate vector
9	τ_{pole}	Angle of one pole	41	C	Zhukovsky line coefficient

Table A1. Cont.

No.	Variable	Description	No.	Variable	Description
10	α_1	First rotation angle in example figure	42	θ_E	Line starting point angular coordinates
11	$E_{1..k \text{ temp}}$	Initial construction points	43	$x_{E_{in}}$	Inner line starting point horizontal coordinates
12	$\Delta\theta_r$	Available angular space	44	$y_{E_{in}}$	Inner line starting point vertical coordinates
13	$w_{bb \ 1..k}$	Barrier bridge vector	45	x_0	Circular barrier center coordinate vector
14	E_n	Notch line starting point	46	R_r	Circular barrier radius vector
15	θ_{notch}	Notch line starting point angular coordinate	47	e_{min}	Minimal eccentricity vector
16	E_{in}	Inner barrier line starting point vector	48	r_E	Line starting point radial coordinate vector
17	E_{out}	Outer barrier line starting point vector	49	x_E	Line starting point angular coordinate vector
18	$\theta_{1..k \text{ temp}}$	Initial barrier construction angular coord.	50	e	Eccentricity vector
19	r_{in}	Inner barrier line starting point radial coord.	51	x_d	Left directrix of hyperbola
20	r_{out}	Outer barrier line starting point radial coord.	52	u	Horizontal w -plane coordinate vector
21	θ_{in}	Inner barrier line starting point angular coord.	53	v	Vertical w -plane coordinate vector
22	θ_{out}	Outer barrier line starting point angular coord.	54	X_{Zhk}	z -plane Zh horizontal vertex vector
23	x_{in}	Inner barrier line intersection point vector	55	Y_{Zhk}	z -plane Zh vertical vertex vector
24	x_{out}	Outer barrier line intersection point vector	56	U_{Zhk}	w -plane Zh horizontal vertex vector
25	X_{in}	Inner barrier line horizontal vertex vector	57	V_{Zhk}	w -plane Zh vertical vertex vector
26	Y_{in}	Inner barrier line vertical vertex vector	58	u_{Zhk}	w -plane Zh intersections vector
27	X_{out}	Outer barrier line horizontal vertex vector	59	u_M	w -plane shaft limit
28	Y_{out}	Outer barrier line vertical vertex vector	60	u_{in}	w -plane MZHV inner barrier intersections
29	X_n	Notch horizontal vertex vector	61	u_{out}	w -plane MZHV outer barrier intersections
30	Y_n	Notch vertical vertex vector	62	T	Period vector of MZHV cosine offset
31	$r_{1..k_{in}}$	Inner barrier fillet vector	63	f	Frequency vector of MZHV cosine offset
32	$r_{1..k_{out}}$	Outer barrier fillet vector	64	Δ_{Dpth}	Barrier depth offset maximum vector
33	α_F	Final rotation angle	65	Δ	MZHV Barrier depth offset vector

References

- Bourzac, K. The Rare-Earth Crisis. *Technol. Rev.* **2011**, *114*, 58–63.
- Justin, R. Rare earths: Neither rare, nor earths. *BBC News*, 23 March 2014.
- Hong, H.S.; Liu, H.C.; Cho, S.Y.; Lee, J.; Jin, C.S. Design of High-End Synchronous Reluctance Motor Using 3-D Printing Technology. *IEEE Trans. Magn.* **2017**, *53*, 4–8. [\[CrossRef\]](#)
- Yamashita, Y.; Okamoto, Y. Design Optimization of Synchronous Reluctance Motor for Reducing Iron Loss and Improving Torque Characteristics Using Topology Optimization Based on the Level-Set Method. *IEEE Trans. Magn.* **2020**, *56*, 36–39. [\[CrossRef\]](#)
- Ban, B.; Stipetić, S. Electric Multipurpose Vehicle Power Take-Off: Overview, Load Cycles and Actuation via Synchronous Reluctance Machine. In Proceedings of the 2019 International Aegean Conference on Electrical Machines and Power Electronics (ACEMP) & 2019 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM), Istanbul, Turkey, 27–29 August 2019.
- Ban, B.; Stipetić, S. Design and optimization of synchronous reluctance machine for actuation of electric multi-purpose vehicle power take-off. In Proceedings of the 2020 International Conference on Electrical Machines (ICEM), Göteborg, Sweden, 23–26 August 2020; pp. 1750–1757.
- Ban, B.; Stipetić, S.; Jercic, T. Minimum set of rotor parameters for synchronous reluctance machine and improved optimization convergence via forced rotor barrier feasibility. *Energies* **2021**, *14*, 2744. [\[CrossRef\]](#)
- Hubert, T.; Reinlein, M.; Kremser, A.; Herzog, H.G. Torque ripple minimization of reluctance synchronous machines by continuous and discrete rotor skewing. In Proceedings of the 2015 5th International Electric Drives Production Conference (EDPC), Nürnberg, Germany, 15–16 September 2015.
- Ferrari, S. Design, Analysis and Testing Procedures for Synchronous Reluctance and Permanent Magnet Machines. Doctoral Thesis, Politecnico di Torino, Torino, Italy, 2020.
- Stipetić, S.; Zarko, D.; Cavar, N. Design Methodology for Series of IE4/IE5 Synchronous Reluctance Motors Based on Radial Scaling. In Proceedings of the 2018 23rd International Conference on Electrical Machines (ICEM), Alexandroupoli, Greece, 3–6 September 2018; pp. 146–151.
- Mirazimi, M.S.; Kiyomarsi, A. Magnetic Field Analysis of Multi-Flux-Barrier Interior Permanent-Magnet Motors Through Conformal Mapping. *IEEE Trans. Magn.* **2017**, *53*, 7002512. [\[CrossRef\]](#)
- Mirazimi, M.S.; Kiyomarsi, A. Magnetic Field Analysis of SynRel and PMASynRel Machines with Hyperbolic Flux Barriers Using Conformal Mapping. *IEEE Trans. Transp. Electr.* **2020**, *6*, 52–61. [\[CrossRef\]](#)
- Taghavi, S.; Pillay, P. Design aspects of a 50hp 6-pole synchronous reluctance motor for electrified powertrain applications. In Proceedings of the IECON 2017—43rd Annual Conference of the IEEE Industrial Electronics Society, Beijing, China, 29 October–1 November 2017; pp. 2252–2257.
- Gamba, M.; Pellegrino, G.; Cupertino, F. Optimal number of rotor parameters for the automatic design of Synchronous Reluctance machines. In Proceedings of the 2014 International Conference on Electrical Machines (ICEM 2014), Berlin, Germany, 2–5 September 2014; pp. 1334–1340.

15. Cupertino, F.; Pellegrino, G.; Cagnetta, P.; Ferrari, S.; Perta, M. SyRE: Synchronous Reluctance (Machines)—Evolution. Available online: <https://sourceforge.net/projects/syr-e/> (accessed on 11 March 2021).
16. Zarko, D. A Systematic Approach To Optimized Design of Permanent Magnet Motors with Reduced Torque Pulsations. Ph.D. Thesis, University Of Wisconsin-Madison, Madison, WI, USA, 2004.
17. Weisstein, E.W. Conformal Mapping. Available online: <https://mathworld.wolfram.com/ConformalMapping.html> (accessed on 11 March 2021).
18. Olver, P.J. Complex Analysis and Conformal Mapping, Chapter 6. In *Complex Analysis and Conformal Mapping*; University of Minnesota: Minneapolis, MN, USA, 2011.
19. Pellegrino, G.; Cupertino, F.; Gerada, C. Automatic Design of Synchronous Reluctance Motors Focusing on Barrier Shape Optimization. *IEEE Trans. Ind. Appl.* **2015**, *51*, 1465–1474. [[CrossRef](#)]
20. Lu, C.; Ferrari, S.; Pellegrino, G. Two Design Procedures for PM Synchronous Machines for Electric Powertrains. *IEEE Trans. Transp. Electrification* **2017**, *3*, 98–107. [[CrossRef](#)]
21. Ban, B.; Stipetić, S. Systematic Metamodel-based Optimization Study of Synchronous Reluctance Machine Rotor Barrier Topologies. *Struct. Multidiscipl. Optim.* **2022**, *in review*.