Explicit solution to the 3-D contact analysis problem of non-conjugate gear teeth in parallel-axis and cross-axis helical configurations

Sabit Kurmashev, BSc Mechanical Engineering

Submitted in fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering

School of engineering
Department of Mechanical Engineering
Nazarbayev University

53 Kabanbay Batyr Avenue,
Astana, Kazakhstan, 010000

Chief Supervisor: Christos Spitas

December 13, 2018

DECLARATION

I hereby, declare that this manuscript, entitled “Explicit solution to the 3-D contact analysis problem of non-conjugate gear teeth in parallel-axis and cross-axis helical configurations”, is the result of my own work except for quotations and citations which have been duly acknowledged. I also declare that, to the best of my knowledge and belief, it has not been previously or concurrentl submitted, in whole or in part, for any other degree or diploma at Nazarbayev University or any other national or international institution.

Name: Sabit Kurmashev

Date: 13.12.2018

Abstract

Modern gear tooth contact models are implicit multiple equation systems which rely on numerical solution techniques. Often, they require initial guess values. If they are not accurate enough, problems with solution convergence may appear and demand significant computational effort to be solved. In this paper a new model is developed using a modified version of the fundamental law of gearing in three dimensions. Gear surfaces were parametrized using two parameters which are radius from center of gear to the point of contact and its projection on the center line. Due to such specific parametrization, it was possible to derive a set of modified vector equations for surface coincidence and tangency which could be reduced to 2 scalar explicit equations with 2 unknowns, and which provides effectiveness in finding the solution of tooth contact compared to other models. The profile tangency conditions are expressed through the following aspects: cross product of surface normal that is equal to zero, which is more closely linked to the physical meaning of the tangency condition, and the fact that difference of vectors from centers of gears to point of contact at the meshing position is equal to distance between centers of gears. This formulation allows to eliminate the need for careful determination of initial values for the numerical solution Tooth contact analysis was developed on Wolfram Mathematica based on the two involute helical gears in parallel axis and

cross axis configurations. Then surface modifications were applied to create non-conjugate contact in the both configurations. The model was compared to the commercial analog based on the accuracy and computational power.

Table of Contents

List of Abbreviations

a_{12}	gears' center distance
$B1$	helix angle of helical gear
f	vector function of parametric gear tooth surface
n	normal vector function of gear tooth surface
q	quaternion
R	generic rotary translation matrix
r	rotated vector function of gear tooth surface
rg	base radius of helical gear
T	theta angle of helical gear
u	first parameter
v	second parameter
zn	coordinate of first side end of gear along axis of rotation
zx	coordinate of second side end of gear along axis of rotation
w	axis of rotation
α	alpha angle of helical gear
θ	angle of rotation

List of Figures

List of Tables

Chapter 1. Introduction

1.1 General

Gears are one of the most important devices for torque transmission in engineering field. Using such phenomenon as gear advantage, or transmission ratio, they can be used to change the velocity, torque or direction of rotation of power supply. There are several types of gears: spur, bevel, helical, cycloidal and others.

Helical gears are one of the quietest among them and usually are used in high-speed or high-power transmission application, or where noise cannot be ignored. The teeth of helical gears are cut under angle comparing to straight spur gear teeth. The final shape is helix which contributes to smooth and quite meshing. Moreover, simultaneously more than one pair of teeth is in contact, which contributes to gradual load distribution among them, resulting in less vibrations, absence of shock loads and less wear. The main disadvantage of this type of gear is that it always experiences axial forces and good thrust bearings are required for steady operation. [1]

1.2 Literature Review

Computer simulation of gear tooth contact is an important part of gear design. It allows to foresee all effects of meshing during design stage which can include misalignment and transmission errors, main causes of noise, vibrations and tooth damage. Thus, simulation of gear meshing led to the development of gears with involute profiles which allowed to maintain constant velocity ratio transmission. This area of gear theory is well studied and established by Faydor L. Litvin [2], who developed so called Tooth Contact Analysis (TCA) and Local Synthesis, along with such researchers as Kai, Baxter, Gleason and others [1]. Litvin's model computes contact between two teeth for a given machine tool settings. However, the main drawback of the model is that it requires guessing of location of point of contact to obtain exact solution which is problematic as causes necessity of substantial computational power or knowing the solution before.

The method was used for TCA and stress analysis of spiral bevel gears meshing [3, 4]. The procedure included Local Synthesis, or in other words calculation of machine tool settings after which contact line could be found and modified in order to decrease stress and transmission errors in the region using analytical approach. The equation derived was so efficient in computing the contact of involute profile gears that F.

Litvin adapted this solution to different types of envelope surface gears such as worm [5], [6], helical, spur and cycloidal.

Practically involute gears are not always used due to several reasons. The profile of involute gear may be intentionally modified, for example, in order to change stress field in the tooth. Another reason may be simply due to errors produced during manufacturing stage. Furthermore, some gears, e.g. Geneva and Wildhaber-Novikov, are inherently not related to conjugate action gears. The theory, developed by Litvin and others does not solve such kind of problems with the same effectivity as for conjugate gears [1].

Various models were proposed on non-conjugate action gears. One of them was developed by Zhang and Fang [7] who aimed to create a model for tooth meshing of helical gears with deformed surfaces which were designed to lower transmission errors and accounted manufacturing errors and specific load distribution. Applying this methodology, authors were able to improve meshing quality by accounting load distribution and modifying it.

K. Mao [8] in his research paper computes the tooth contact model that can be applied to reduce gear surface wear. Using conventional tooth contact theory, 3D model and FEA programs he was able to find the path of contact, accounting surface evolution due to wear. By employing

micro-geometry modifications throughout the simulation of whole working cycle, FEA could predict accurately fatigue wear. Based on the method, Mao was able to optimize the contact of gear tooth reducing fatigue wear and experimentally prove it.

Another research on wear prediction and modification of gear profile is performed by Bajpai et al [9] for spur and helical gears. The basis of the method is Archard's wear formulation that is used along with finite element contact mechanics method to find contact path. The suggested methodology allows to perform iterative numerical simulation of gear surface modification imitating wear mechanism at the contact path. The method is flexible, can be used with finite element program and allows to import any gear geometry for computations. Later the model was validated experimentally and was proved to be efficient in wear prediction.

The method developed by Kolivand and Kahraman [10] allows to perform tooth contact analysis for spiral bevel and hypoid gears with profile modifications obtained due to manufacturing errors and fatigue wear. It is based on so called "ease-off topography" which accounts the error of mating of real gear surface with the conjugate of real pinion surface. Such feature allows to reduce computational power demand, speed up solution time and avoid extra interpolations of solution. The

topography is calculated using machine tool settings, model dimensions of the gears and manufacturing error and surface wear values. Then, unloaded and loaded contact is found semi-analytically. The method was applied to spiral bevel and hypoid gears and results showed that the methodology is effective.

Some models were not confined by only one pair of teeth, Parker et al. [11] in their work also tried to integrate conventional tooth contact theory in finite element analysis for spur gears resulting in so called finite element/contact mechanics model. They studied the dynamic response of gears under different torques considering non-uniform motion of gears. In comparison to other methods which utilize meshing of two or three teeth, this model considered meshing of the whole gear and virtually rotated it throughout the whole cycle of rotation. This allowed to directly calculate all dynamic forces and the moments of contact loss. The model was simulated on planetary gears and successfully verified by experiments.

The work presented in this paper is based on paper written by C. Spitas and V. Spitas [12]. They developed 2D analytical model of non-conjugate gear tooth contact which have better solution in terms of simplicity and convergence compared to Litvin's one. They computed equations for non-aligned tooth contact of gears with modified surface

which accounted manufacturing error. Basis for numerical computation are equations for tangency of gear tooth which were modified for generated modified gear profile and which allowed to calculate the position of second gear. The advantage of the method was that model could be reduced to one equation using analytical approach. The method was compared to Litvin's solution based on the number of iterations and time required to solve the system. It was found that method suggested by authors required both less time and iterations to compute solution.

Later they [13] presented modified explicit solution for meshing non-conjugate gears in two dimensions which could be solved using given unconditionally stable numerical approach. Again, one-equation solution was analytically derived from rule of gear profile tangency which dramatically simplified computation and provided better convergence. The results produced by the model, which was applied to parametric surface of Geneva gear, demonstrated that solution can be obtained in a fast manner using small amount of iterations.

1.3 Aims & Objectives

The main objective of the paper is to develop analytical model which exploits direct parametrization of gear for 3D meshing of non-conjugate parallel and crossed axes helical gear that satisfy profile tangency condition when surface normals coincide at the point of

contact. The parametrization is based on the radius from center to the point of contact and its projection on the center line parameters which allows one to significantly simplify the problem and use two explicit scalar equations with two unknowns avoiding initial value uncertainties. Thus, the simulated model will have better solution in terms of computational power compared to Litvin's solution, who, on the other hand, exploits conventional parametrization using Cartesian coordinate system and model, requiring guessing of initial state of contact.

Chapter 2. Methodology

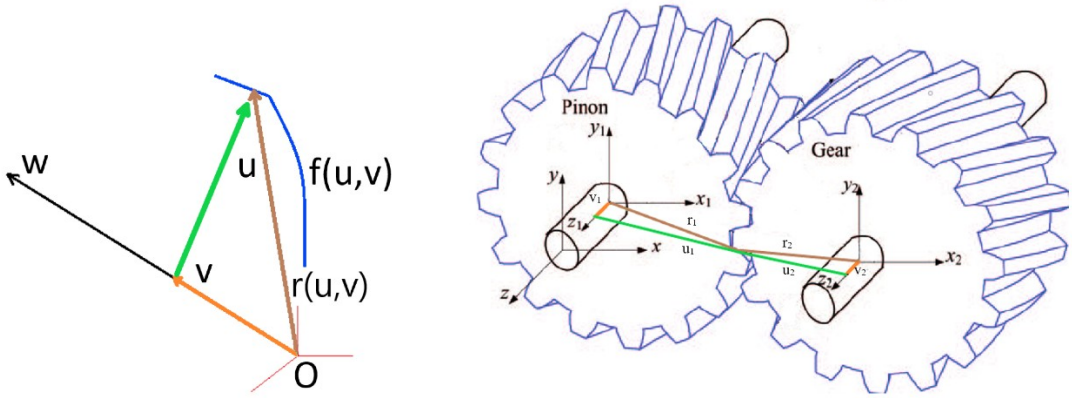
2.1. Parametrization of tooth surface

Gear tooth surface is mapped as follows:

$$p: R^2 \rightarrow R^3$$

Gear surfaces are parametrized using two parameters u and v which are radius from center of gear to the point of contact and its projection on the center line, respectively. Due to such specific parametrization, it is possible to derive a set of modified vector equations for surface coincidence and tangency which can be reduced to 2 scalar explicit equations with 2 unknowns. Both are parameters are visually shown on the following figure, where green line represents u parameter and orange – v. r is the surface vector

Figure 1. Proposed direct parametrization of tooth surface: schematically (left) and on the contacting gears (right).



For the left picture, w s a center of rotation of gear, f is a surface of tooth, r is a surface vector which is based on two parameters u and v. For the right picture, x, y and z are global coordinates. x_1, y_1 , z_1 and x_2, y_2 , z_2 are coordinate systems of pinion and gear, respectively.

2.2. Explicit solution

The surface of first gear is expressed as $f_1(u_1, v_1)$ which C1-continuous, revolves around axis w_1 by an angle θ_1 (see Figure 1). The global position of contact is represented by vector $r_1(u_1, v_1)$ and surface normal vector is $n_1(u_1, v_1)$. Similarly, surface $f_2(u_2, v_2)$ is C1-continuous, revolves around axis w_2 by an angle θ_2 . The global position of contact is $r_2(u_2, v_2)$ and its normal vector is n_2 and is in

non-conjugate contact with surface $f_1(u_1, v_1)$. The vector connecting the origins of the two bodies, or center distance, is a_{12} (see Figure 2).

Figure 2. 2D schematic representation of gear tooth rotation.

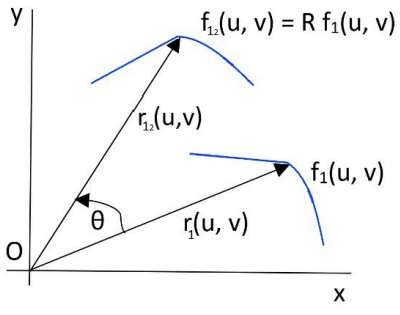
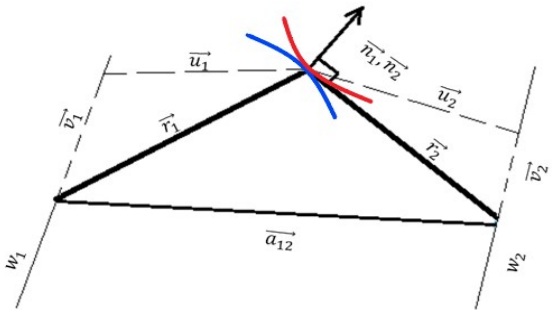


Figure 3. Schematic representation of tooth contact. Blue and red lines represent two gear tooth surfaces.



The profile tangency condition is expressed in terms of cross product of surface normal that is equal to zero, which is more closely linked to the physical meaning of the tangency condition (1), and the fact that vector difference $r_1 - r_2$ at the position of contact is equal to distance between centers of gears (2):

$$n_1(u_1, v_1) \times n_2(u_2, v_2) = 0 \quad (2.1)$$

$$r_1(u_1, v_1) - a_{12} - r_2(u_2, v_2) = 0 \quad (2.2)$$

Each surface normal from equation (1) can be expressed in terms of cross product of profile tangents:

$$n_1(u_1,v_1)=\frac{\partial r_1(u_1,v_1)}{\partial u_1}\times\frac{\partial r_1(u_1,v_1)}{\partial v_1} \quad (2.3)$$

$$n_2(u_2,v_2)=\frac{\partial r_2(u_2,v_2)}{\partial u_2}\times\frac{\partial r_2(u_2,v_2)}{\partial v_2} \quad (2.4)$$

Here vector r_1 is rotation of surface vector f_1 . It can be performed using rotation matrix R .

$$r_1=R_1(\theta_1)f_1(u_1,v_1) \quad (2.5)$$

$$r_2=R_2(\theta_2)f_2(u_2,v_2) \quad (2.6)$$

At this point, we have 4 unknowns: u_1,v_1 , u_2 and v_2 . Angle θ_2 is a function of u_2 and v_2 . Parameters u_2 and v_2 can be found easily due to specific direct parametrization from the geometry (see Figure 2):

$$v_2=proj_{w_2}r_2(u_2,v_2)=proj_{w_2}f_2(u_2,v_2) \quad (2.7)$$

$$v_2=proj_{w_2}(r_1(u_1,v_1)-a_{12})=w_2\cdot(r_1(u_1,v_1)-a_{12})=F_v(u_1,v_1) \quad (2.8)$$

From Pythagorean Theorem:

$$u_2^2+v_2^2=r_2(u_2,v_2)^2=f_2(u_2,v_2)^2 \qquad (2.9)$$

$$u_2=\sqrt{r_2(u_2,v_2)^2-v_2^2}=\sqrt{f_2(u_2,v_2)^2-v_2^2} \qquad (2.10)$$

$$u_2=\sqrt{(r_1(u_1,v_1)-a_{12})^2-v_2^2}=F_u(u_1,v_1) \qquad (2.11)$$

The angle θ_2 for rotation can be calculated from dot product:

$$f_2(F_u,F_v)\cdot r_2(F_u,F_v)=r_2(F_u,F_v)^2\cos\theta_2 \qquad (2.12)$$

Here, $f_2(F_u,F_v)$ is the surface vector to the point of contact when gear is not rotated to the meshing position with the second gear. Hence, θ_2 can be found as:

$$\theta_2=\cos^{-1}\frac{f_2(F_u,F_v)\cdot r_2(F_u,F_v)}{r_2(F_u,F_v)^2}=\cos^{-1}\frac{f_2(F_u,F_v)\cdot(r_1(u_1,v_1)-a_{12})}{r_2(F_u,F_v)^2} \qquad (2.13)$$

The sign of angle θ_2 can be found from the dot of cross product of vectors f and r:

$$\frac{f_2(F_u,F_v)\times r_2(F_u,F_v)}{\left\|f_2(F_u,F_v)\times r_2(F_u,F_v)\right\|}\cdot w_2=\pm 1 \qquad (2.14)$$

For a clockwise rotation it will be positive and negative for anticlockwise. Gathering equations together, we have:

$$\theta_2=\frac{f_2(F_u,F_v)\times(r_1(u_1,v_1)-a_{12})}{r_2(F_u,F_v)^2}\cdot w_2\cos^{-1}\frac{f_2(F_u,F_v)\cdot(r_1(u_1,v_1)-a_{12})}{r_2(F_u,F_v)^2}$$

$$¿F_\theta(u_1,v_1,F_u,F_v) \qquad (2.15)$$

All terms are expressed in terms of two parameters: u_1 and v_1 . Finally, equation 2.7 can be rewritten as:

$$R_1(\theta_1)\left(\frac{\partial f_1(u_1,v_1)}{\partial u_1}\times\frac{\partial f_1(u_1,v_1)}{\partial v_1}\right)\times R_2(F_\theta)\left(\frac{\partial f_2(F_u,F_v)}{\partial u_1}\frac{\partial u_1}{\partial u_2}\times\frac{\partial f_2(F_u,F_v)}{\partial v_1}\frac{\partial v_1}{\partial v_2}\right)=0$$

(2.16)

Thus, knowing the orientation and axis of rotation of first body, the orientation of second body with respect to its axis of rotation can be easily calculated. Now, from equation 2.16 the meshing of the gears can be computed from two out of three scalar equations with two unknown parameters.

2.2. Simulation

The model was coded in Wolfram Mathematica 11.3 and compared to the Litvin's solution of TCA as a benchmark. The test runs were performed on a computer with processor Intel(R) Core(TM) i5-4460 with clock frequency 3.2GHz and 8 GB RAM. The Litvin's solution is given in Appendix.

The simulation of two helical gears with similar properties for the sake of simplicity was performed. The parameters are:

Table 1. Gears characteristics

Characteristic	**Gear 1**	**Gear 2**
Module, cm	3	
Number of teeth	30	
Pressure angle, °	20	
Base diameter, cm	95.80572457	
Center distance, cm	105	

The orientation of the first gear was chosen to be 30 degrees for a parallel-axis and 45 degrees for a cross-axis configurations. Surface of tooth itself was rotated by and 90 and 45 degrees clockwise around centers of gears for parallel and cross configurations, respectively.

Table 2. Helix angle and surface orientation in both configurations

Characteristic	**Parallel Axis**	**Cross axis**
Helix angle, °	30	45
Tooth surface orientation, ° (clockwise)	90	45

The parametrization of helical tooth surface was calculated and expressed in terms of given parameters u and v from a general equations for a given machine tool settings. The surface of helical gears was modified by adding extra terms to the equation of involute gear to simulate random error during the manufacturing. The equations are:

Table 3. Comparison between involute and modified parametrization

Involute parametrization	**Modified parametrization**
$x_i = u * \cos\left(\alpha + \frac{\beta * v - zn}{zx - zn}\right)$	$x_m = u * \cos\left(\alpha + \frac{\beta * v - zn}{zx - zn}\right) + 0.001u^2$
$y_i = u * \sin\left(\alpha + \frac{\beta * v - zn}{zx - zn}\right)$	$y_m = u * \sin\left(\alpha + \frac{\beta * v - zn}{zx - zn}\right) + 0.01u$
$z_i = v$	$z_m = v$

Where:

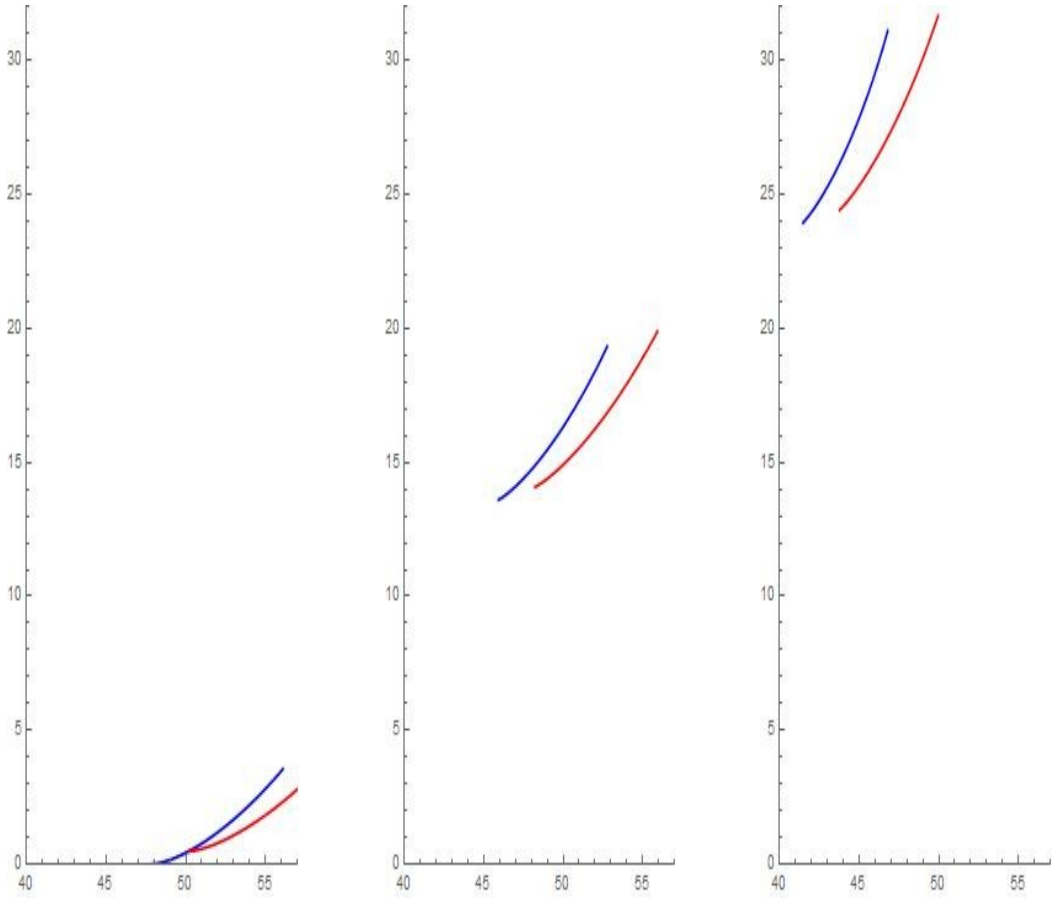
$$T = \sqrt{\left(\frac{u}{rg}\right)^2 - 1}$$

$$\alpha = \cos^{-1}\left(\frac{rg}{u} * (\cos(T) + T * \sin(T))\right)$$

Here, β is a helix angle and rg is a base radius. The extra terms are $0.001u^2$ in x coordinate and $0.01u$ in y coordinate.

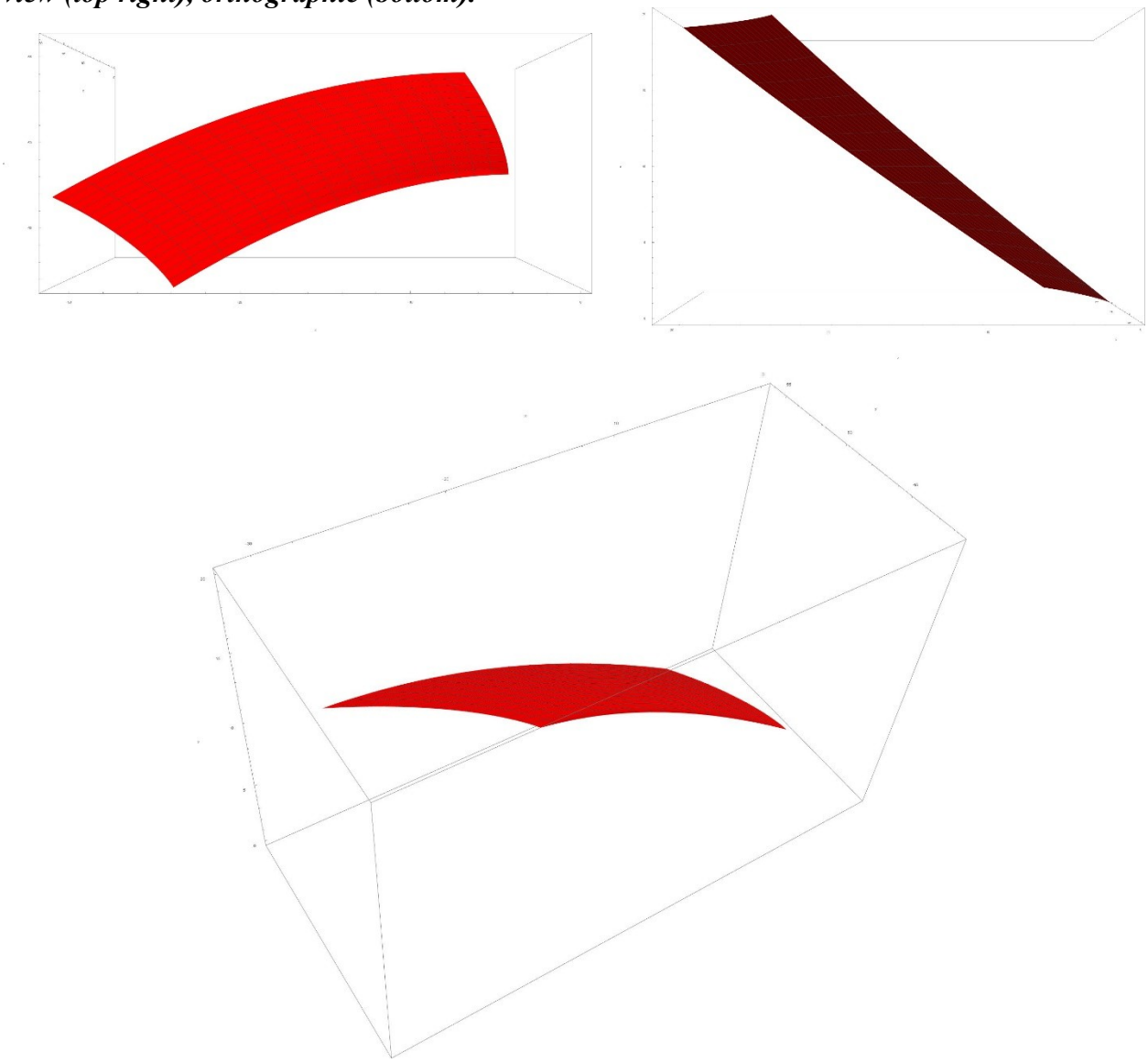
The following figures show the change in profile with increase in parameter v, or in other words, cross section of gear's tooth surface at each position on a centerline of gear.

Figure 4. Gear profile evolution with variation of its position along centerline. Blue- involute, Red – modified for v = 0 (left), 10 (middle) and 20 (right).



Tooth contact analysis was performed on a pair of modified parametric involute gear surfaces using the methodology developed in this work. As it was shown, two gears with similar parameters are modeled (see Table 1). The parametrization of surfaces was done using similar properties for both types of gears.

Figure 5. Parametric surface of helical gear: front view (top left), right view (top right), orthographic (bottom).



The second gear was parametrized using the same parametrization but with negative helix angle. Furthermore, for imitating crossed-axis configuration, x and z coordinates of second gear were changed with each other in order to rotate tooth surface 90 degrees.

The schematic of gear teeth meshing in both configurations are shown on the next figures. Red surface represents fixed given gear and blue – one that is required to be computed.

Figure 6. Parametric gears in parallel configuration.

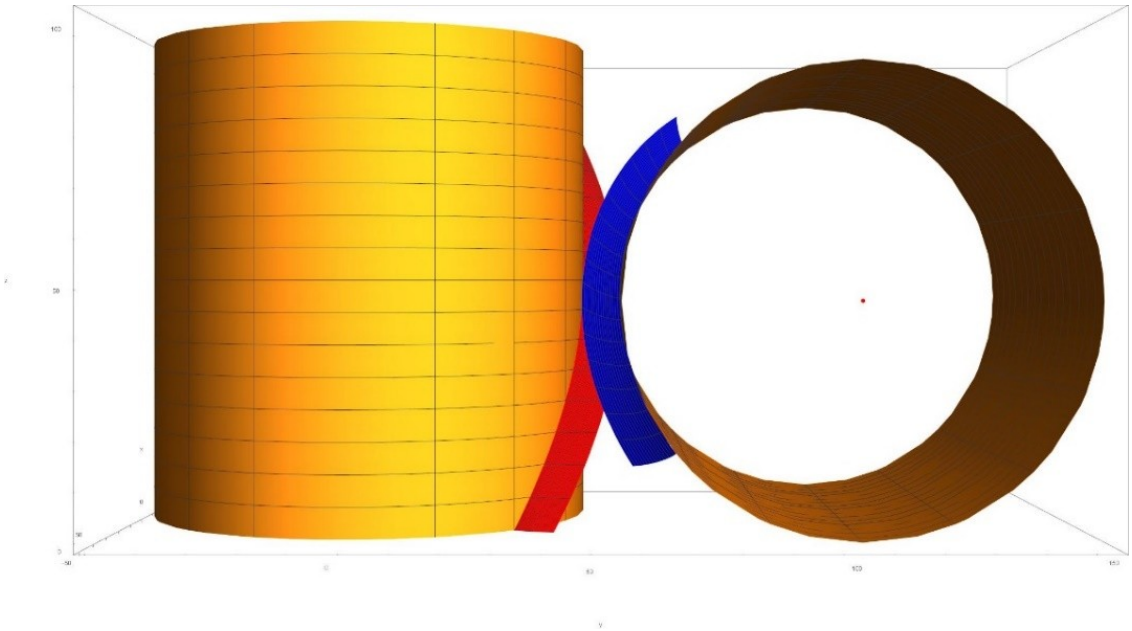
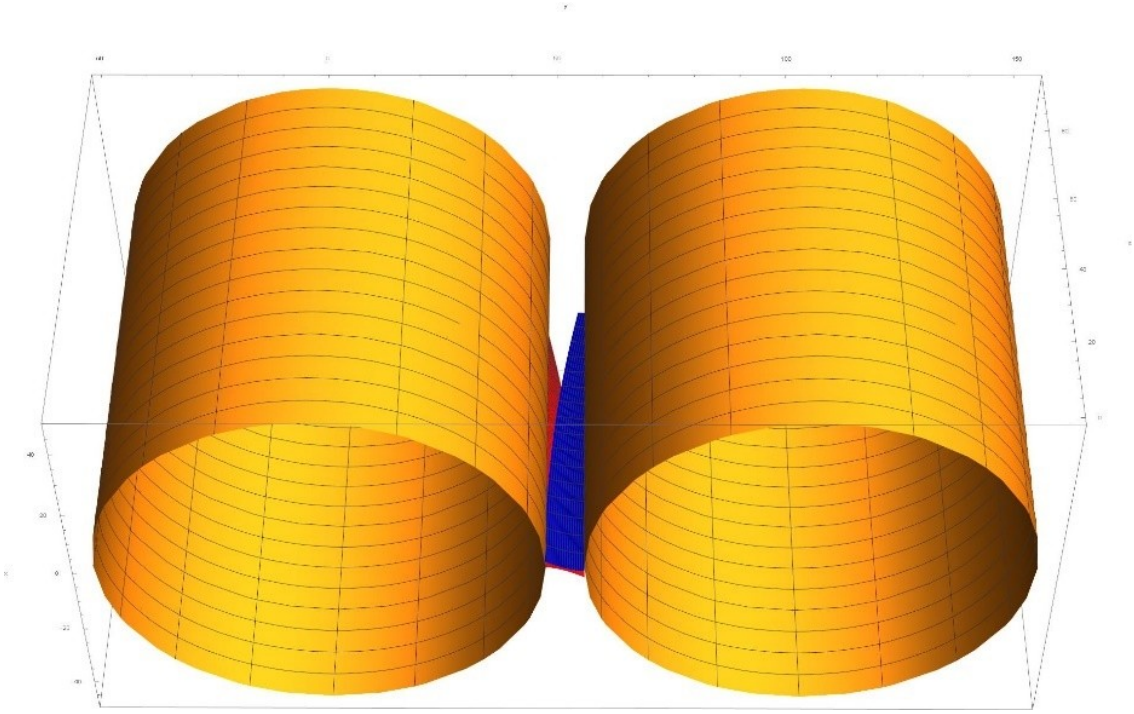


Figure 7. Parametric gears in crossed configuration.



Chapter 3. Results and Discussion

The results will be compared on a basis of the following parameters: accuracy compared with each other, time of calculation and

number of iterations. During the simulation, it was not possible to find the solution of commercial solver without inputting guess values. Commercial model simply did not converge if guess values were far from real ones. They were obtained from the results of computations using proposed model. The numerical calculation was performed using command FindRoot which utilizes Newton-Raphson method by default. The time of calculation was obtained using command Timing. Angular position of tooth surface of leading gear (left gear on Figure 5) was fixed and set to be 90 degrees, rotating from the top clockwise. In the parallel configuration the results are following:

Table 4. Parallel configuration results.

Characteristic	Proposed	Commercial
Time spent (rounded by program), s	0.0720397	0.0253373
Point of contact, (x, y, z)	(-1.43883, 54.0307, 0.)	(-1.43883, 54.0307, 0.)
Number of Iterations	42	1
Time per iteration, s	0.001715	0.02534

The next figures show identical meshing tooth surfaces for both models:

Figure 8. Commercial model solution in parallel configuration.

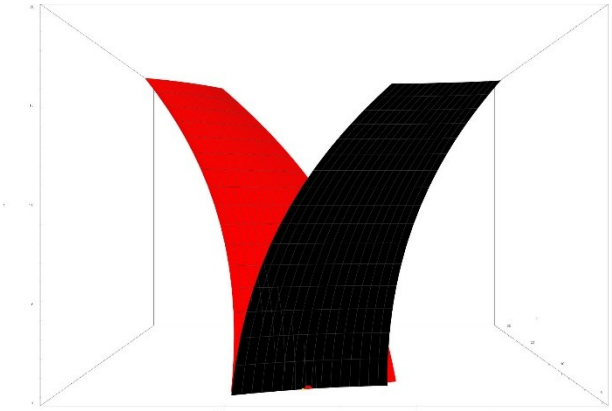
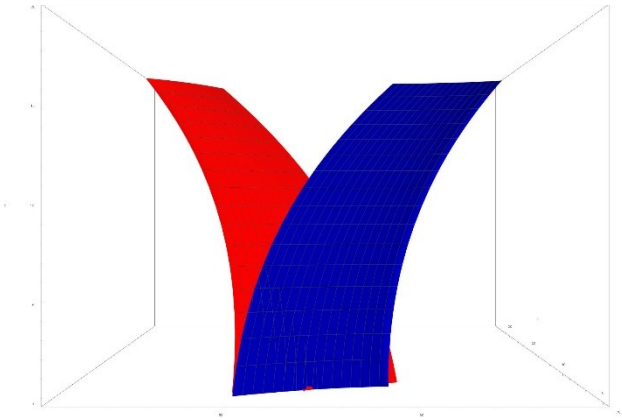


Figure 9. Proposed model solution in parallel configuration



In the crossed-axis configuration, with angular position of tooth surface of leading gear equal to 45 degrees, results are following:

Table 5. Crossed configuration results.

Characteristic	Proposed	Commercial
Time spent (rounded by program), s	0.0844245	0.0288288
Point of contact, (x, y, z)	(-12.8859, 53.6471, 62.146)	(-12.8859, 53.6471, 62.146)
Number of iterations	33	3
Time per iteration, s	0.002558	0.009610

Again, the input guess values for commercial model were obtained from proposed model. The following graphs give visual representation of meshing.

Figure 10. Commercial model solution in crossed configuration.

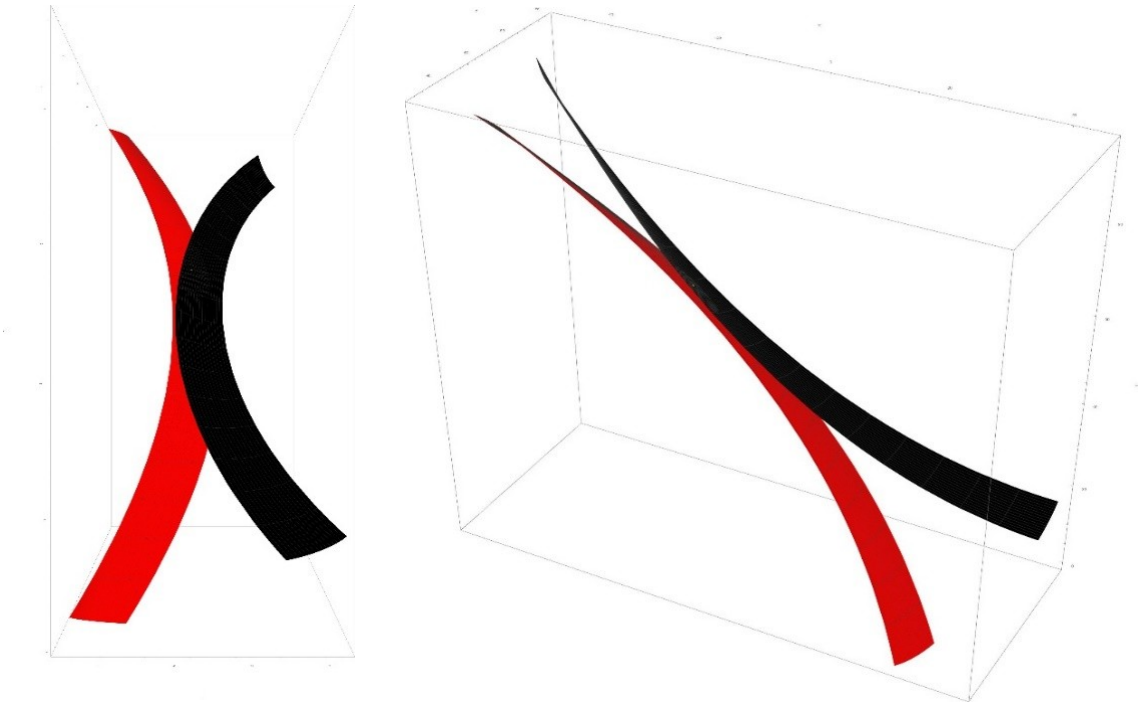
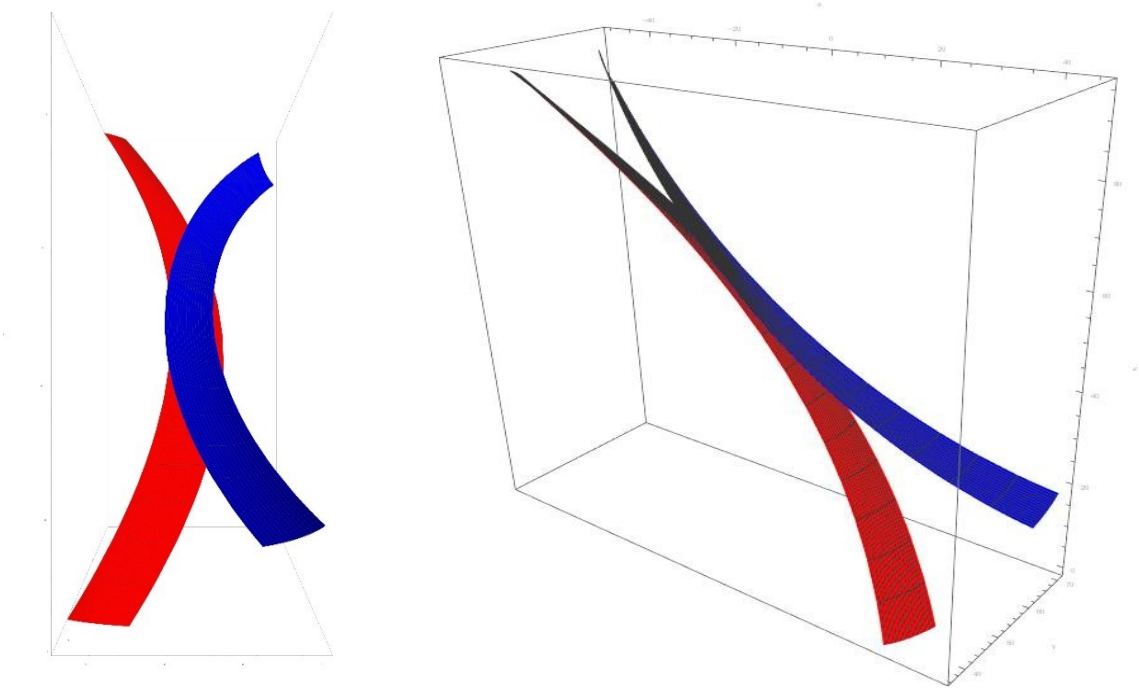


Figure 11. Proposed model solution in crossed configuration.



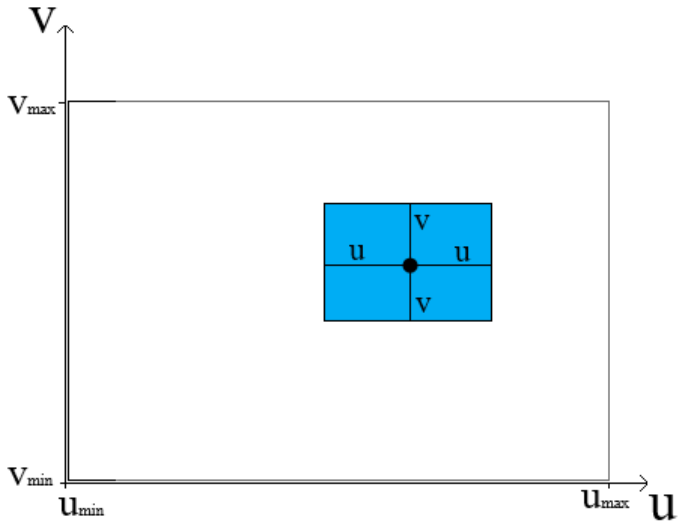
As it can be seen from the graphs, they look identical when both models are compared with each other. As it was mentioned, the model

does not suffer from necessity of knowing point of contact in order to find the solution. On the other hand, commercial model failed to solve the problem when starting point for numerical computation is set far from real contact location, e.g. when point is selected to be at the border of the parametric surface. This model, as it was expected, succeeded only when guess value was close to the real point of contact. However, results also show that proposed method takes almost three times more time to solve the problem. Moreover, there are more number of iterations required to solve the same problem. However, the time spent per iteration shows that increase of time is a result of search routine. Proposed method is more effective than commercial, as later utilizes guess values close to the true ones and necessity in search routine does not affect overall time as it does in former model. Hence, considering time spent per iteration, proposed model is more efficient than commercial one. Finally, as it can be noticed the model as accurate as commercial model and finds the same coordinates of point of contact.

As the time for commercial model was calculated using guess values obtained from the proposed model, it was decided to run several tests with random number generator (RNG). At first, all guess values were generated randomly, that led to fail in convergence when they were

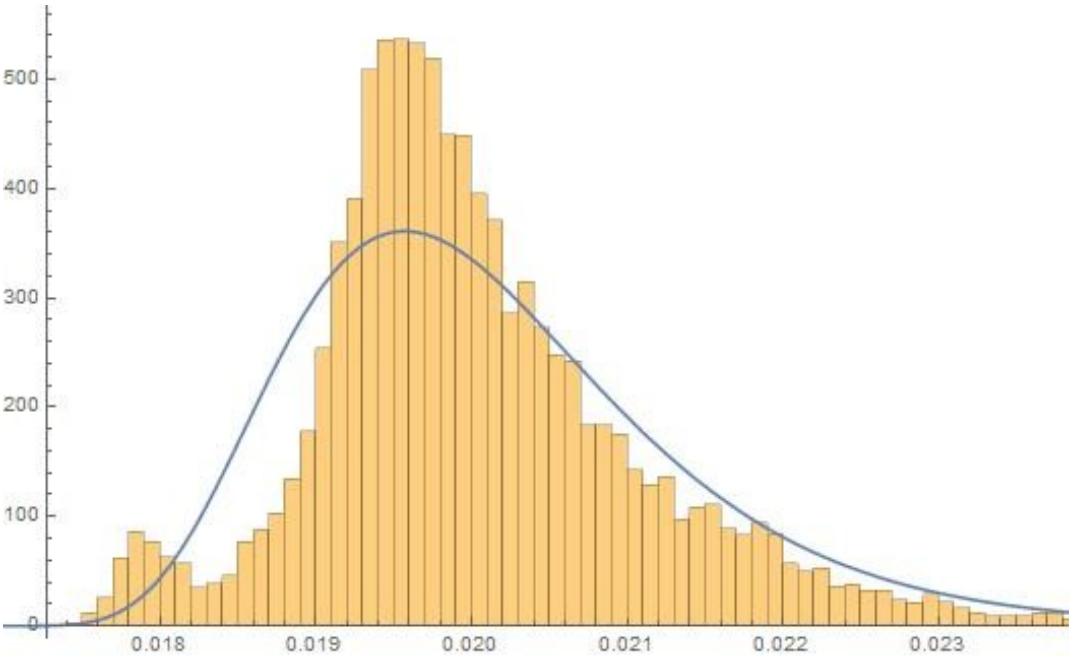
far from close. Hence, RNG was limited to a certain region within one unit from true value.

Figure 12. Search region (blue) on parametric surface of gear tooth.



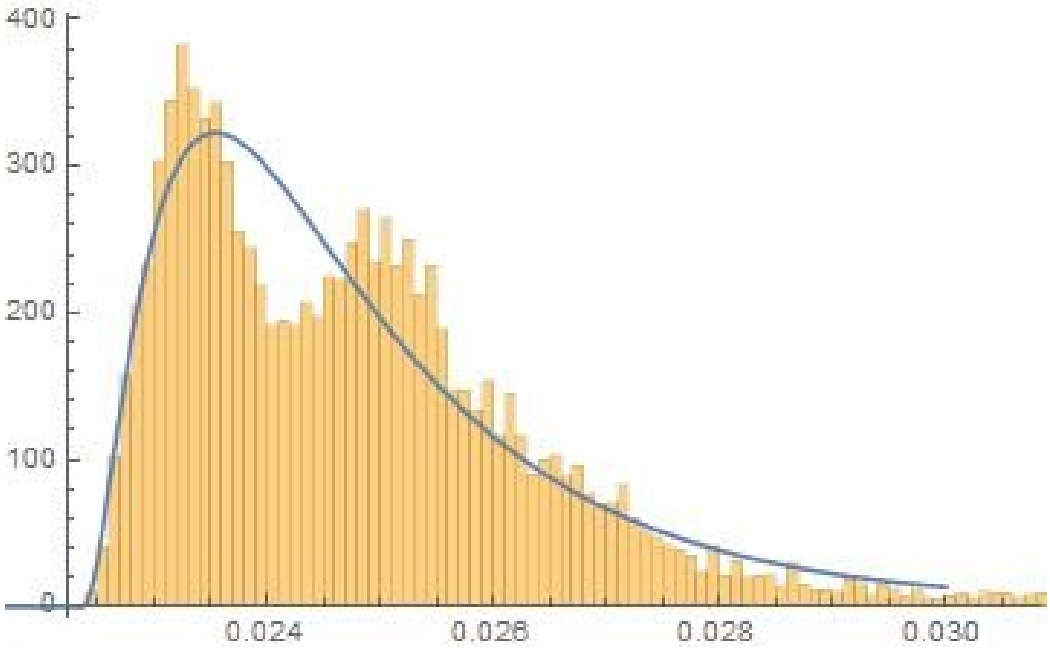
For example, for parallel case both parameters were randomly chosen from the following regions: $50.397<u_1,u_2<52.397$ and $0<v_1,v_2<5$. In result, there is a distribution of computation time required for commercial model based on the 10000 simulations. The mean time of calculation was computed to be equal to about 0.02 sec.

Figure 13. Computational time distribution for parallel case.



For crossed axis case, RNG regions were $51.489<u_1,u_2<53.489$ and $0<v_1,v_2<5$. Calculations were repeated 10000 times that led to following results, showed on the next graph. Mean time was computed to be equal to about 0.24 sec.

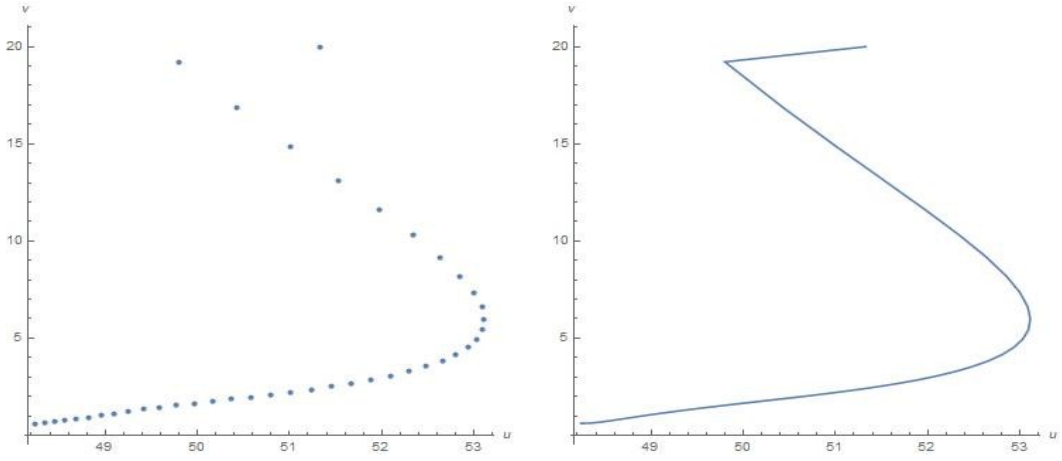
Figure 14. Computational time distribution for crossed case.



Again, it should be noted that most of calculations failed when guess values were far away from true ones and only successful results are shown.

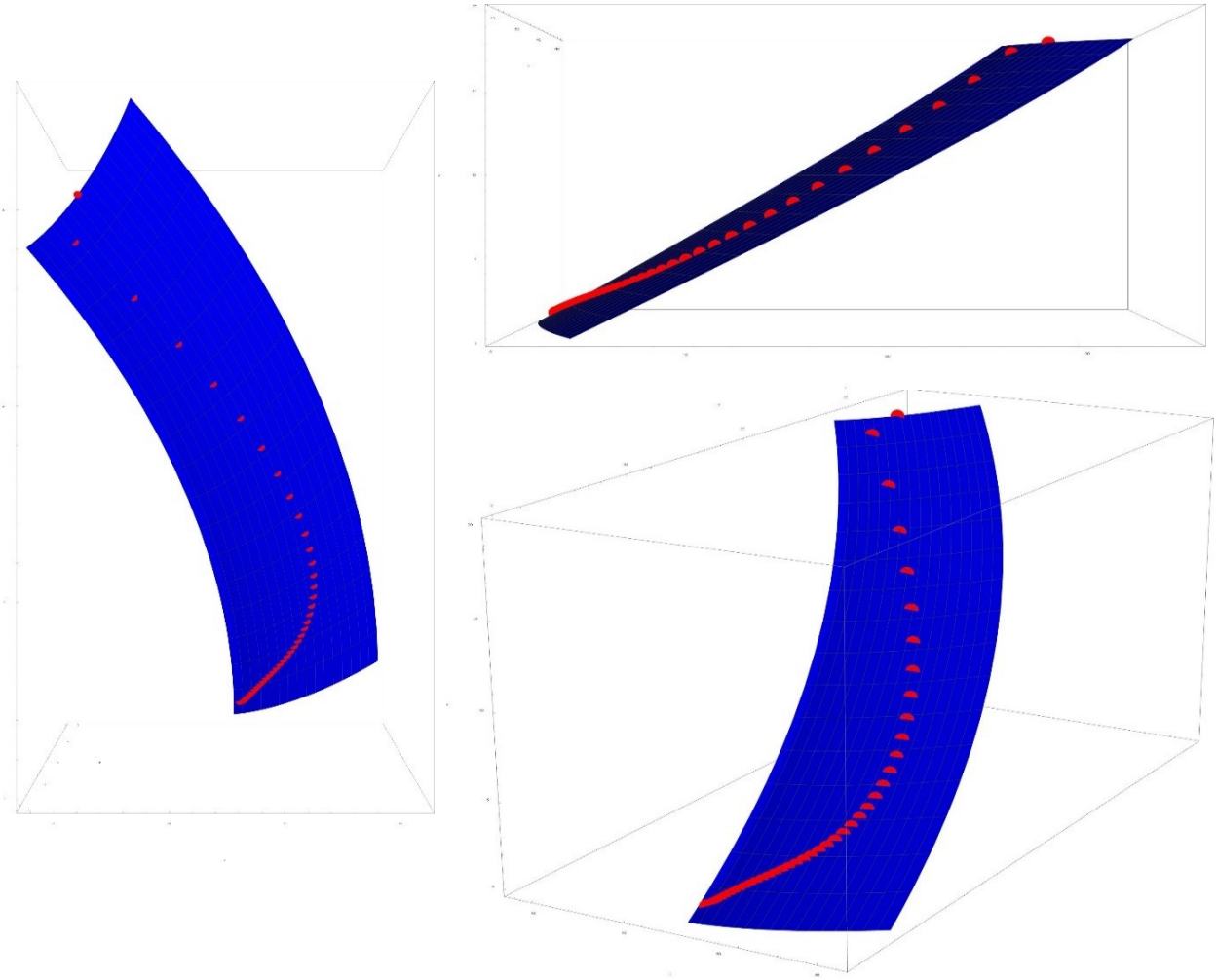
The following graph shows line of contact of the gears in parallel on the parametric surface with parameters u and v for an angular position of leading gear ranging from 67 up to 108 degrees with one-degree step in parallel configuration. The first gear is rotating clockwise.

Figure 15. Line contact in parallel configuration.



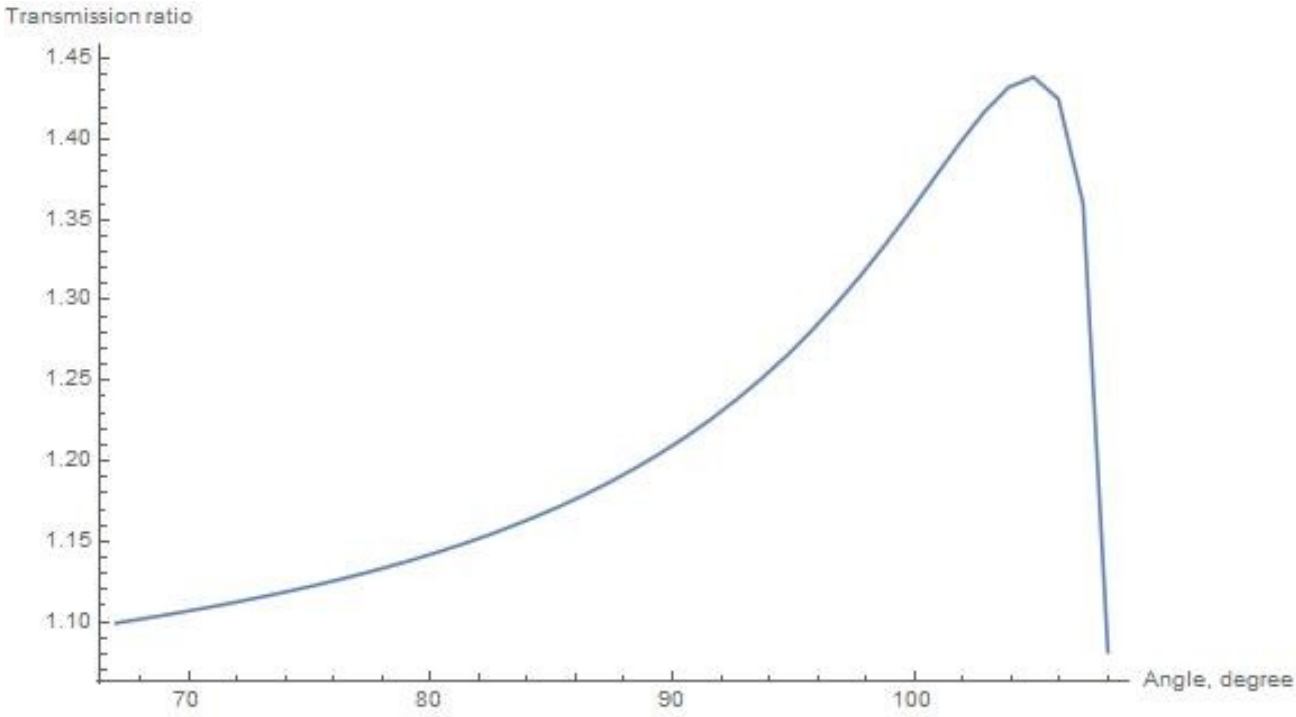
Line contact on the parametric surface is shown in the following picture:

Figure 16. Visual of line of contact in parallel configuration: top view (left), right view (top right) and orthographic view (top bottom)



Finally, transmission error can be calculated based on the code:

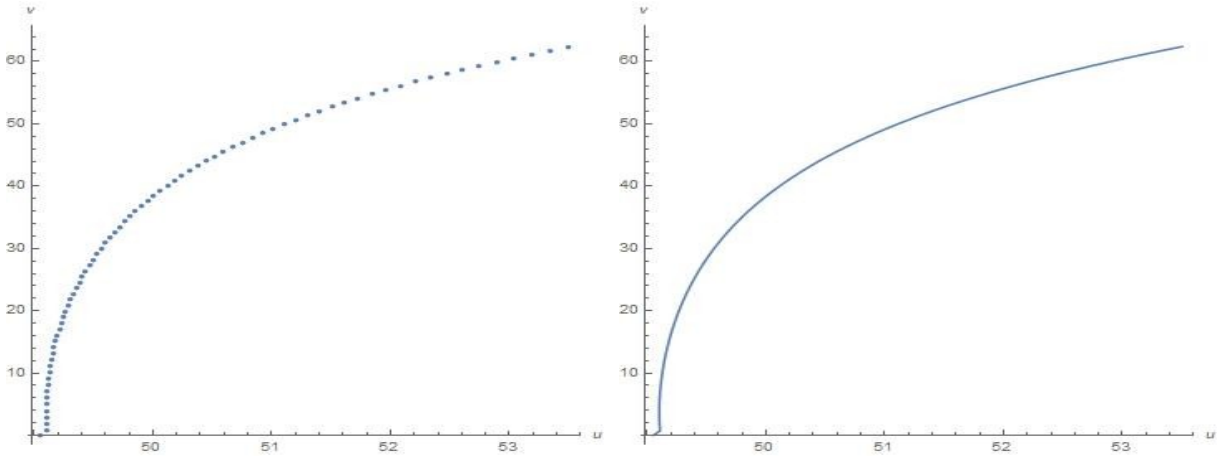
Figure 17. Transmission ratio in parallel configuration.



As it can be seen, due to modified surface transmission ratio is not always uniform and not equal to 1.

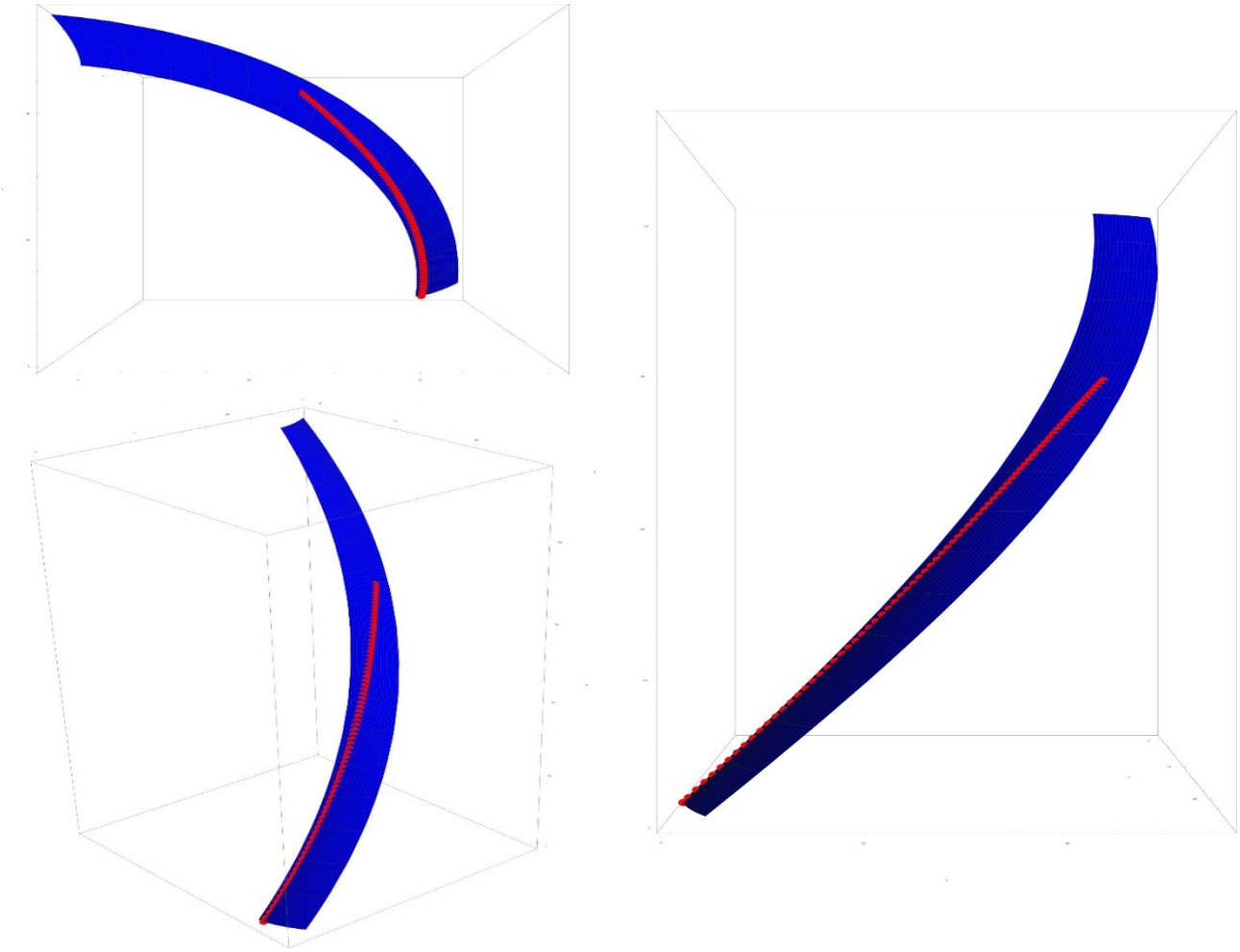
For crossed configuration, line of contact is computed the same way, but for angular positions of the leading gear ranging from 41 to 101 degrees:

Figure 18. Line contact in crossed configuration.



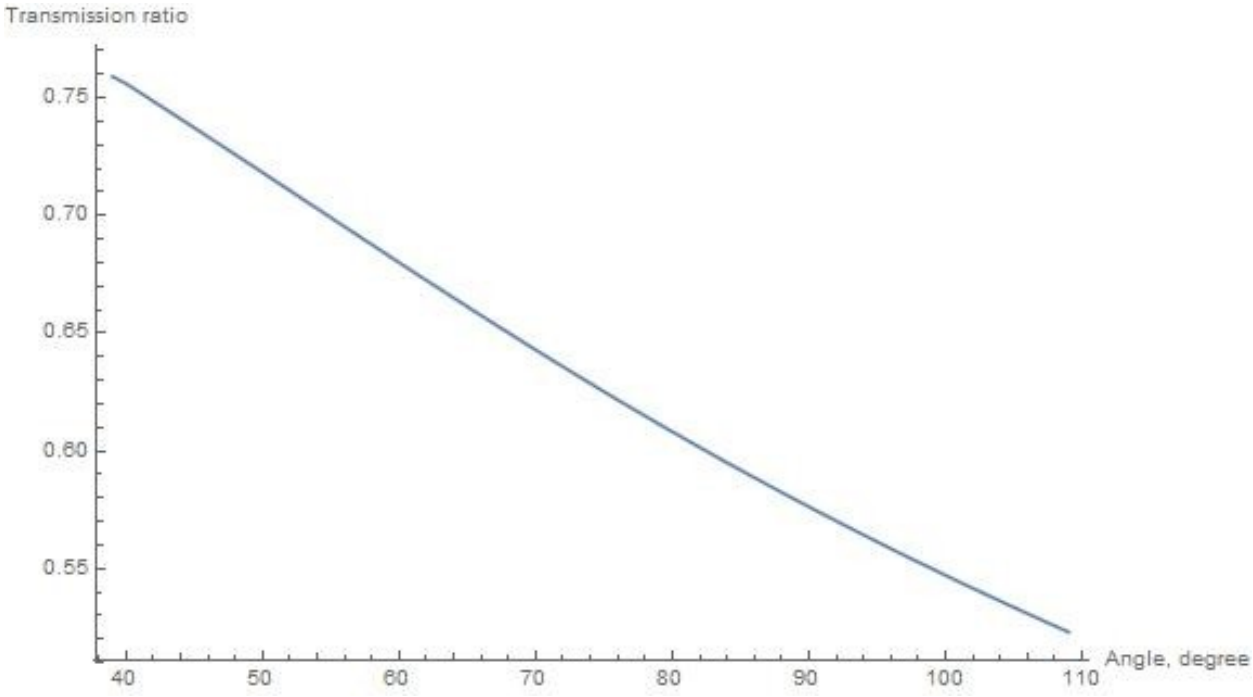
The next figure shows the same line contact on actual surface of the gear's tooth:

Figure 19. Visual of line of contact in crossed configuration: top view (top left), right view (right) and orthographic view (left bottom)



Transmission ratio for crossed configuration, as it was expected, is not uniform and not equal to 1. The main reason, as it was explained, is modified surface, which is not involute.

Figure 20. Transmission ratio in crossed configuration.



The results of line of contact and transmission ratio are identical and, hence, not duplicated and shown here.

Chapter 4. Conclusions

In this paper the new tooth contact analysis model for non-conjugate helical gear meshing was developed from general law of gearing. The methodology allowed to reduce profile tangency relation to two explicit scalar equations with two unknowns, which was computed using Newton-Raphson method. The results revealed that the model is accurate as commercial analog which was chosen to be F. Litvin's solution. Furthermore, it did not require guess values which were vital for mentioned state of the art. On the other hand, the method spent three times more time to solve identical problem which may be caused by

search routine required to solve problem. It took much more iterations for a proposed model and, hence the ratio of time required for calculations was considerably less. Finally, line of contact and transmission ratio for both configurations were computed based on the given model, which were identical to the commercial ones. Several things can be performed for future. Firstly, at this moment code is mostly based on the symbolic data which demands increased computational power compared if it was working with numerical data. Code can be modified. Secondly, the code potentially can calculate contact when two gears are in slight misalignment. Already, it can account small angles of in-line misalignment. Cross configuration is one case of out-of-plane misalignment. Code can be rewritten and optimized to consider in-line, out-of-line and combined misalignments. Finally, developed methodology can be applied to different kinds of gears such as spur, bevel, cycloid and others

Bibliography/References

[1] Litvin, Faydor L., and Alfonso Fuentes. Gear geometry and applied theory. Cambridge University Press, 2004.

[2] Litvin, Faydor L., Galina I. Sheveleva, Daniele Vecchiato, Ignacio Gonzalez-Perez, and Alfonso Fuentes. "Modified approach for tooth contact analysis of gear drives and automatic determination of guess values." Computer Methods in Applied Mechanics and Engineering 194, no. 27 (2005): 2927-2946.

[3] Litvin, Faydor L., and Yi Zhang. Local synthesis and tooth contact analysis of face-milled spiral bevel gears. ILLINOIS UNIV AT CHICAGO CIRCLE, 1991.

[4] Litvin, Faydor L., Alfonso Fuentes, Qi Fan, and Robert F. Handschuh. "Computerized design, simulation of meshing, and contact and stress analysis of face-milled formate generated spiral bevel gears." Mechanism and Machine Theory 37, no. 5 (2002): 441-459.

[5] Litvin, F. L., and C-L. Hsiao. "Computerized simulation of meshing and contact of enveloping gear tooth surfaces." Computer Methods in Applied Mechanics and Engineering 102, no. 3 (1993): 337-366.

[6] Litvin, Faydor L., Ignacio Gonzalez-Perez, Kenji Yukishima, Alfonso Fuentes, and Kenichi Hayasaka. "Design, simulation of meshing, and contact stresses for an improved worm gear drive." Mechanism and Machine Theory 42, no. 8 (2007): 940-959.

[7] Zhang, Yan, and Z. Fang. "Analysis of tooth contact and load distribution of helical gears with crossed axes." Mechanism and Machine Theory 34.1 (1999): 41-57.

[8] Mao, K. "Gear tooth contact analysis and its application in the reduction of fatigue wear." Wear 262.11 (2007): 1281-1288.

[9] Bajpai, P., A. Kahraman, and N. E. Anderson. "A surface wear prediction methodology for parallel-axis gear pairs." Journal of tribology126.3 (2004): 597-605

[10] Kolivand, M., and A. Kahraman. "An ease-off based method for loaded tooth contact analysis of hypoid gears having local and global surface deviations." Journal of Mechanical Design 132.7 (2010): 071004.

[11] Parker, R. G., S. M. Vijayakar, and T. Imajo. "Non-linear dynamic response of a spur gear pair: modelling and experimental comparisons." Journal of Sound and vibration 237.3 (2000): 435-455.

[12] Spitas, C., and V. Spitas. "Direct analytical solution of a modified form of the meshing equations in two dimensions for non-conjugate gear contact." Applied Mathematical Modelling 32, no. 10 (2008): 2162-2171.

[13] Spitas, C., and V. Spitas. "Fast unconditionally stable 2-D analysis of non-conjugate gear contacts using an explicit formulation of the meshing equations." Mechanism and Machine Theory 46.7 (2011): 869-879.

[14]

Appendix

Appendix A. Commercial model

F. Litvin's model [1] was representing commercial analog of proposed model. Profile tangency conditions are following:

$$n_1 - n_2 = 0 \quad (1)$$

$$r_1 - a_{12} - r_2 = 0 \quad (2)$$

Each surface normal from equation (1) can be expressed in terms of cross product of profile tangents:

$$n_1 = \frac{\partial r_1}{\partial u_1} \times \frac{\partial r_1}{\partial v_1} = R_1(u_1, v_1)\left(\frac{\partial f_1(u_1, v_1)}{\partial u_1} \times \frac{\partial f_1(u_1, v_1)}{\partial v_1}\right) \quad (4)$$

$$n_2 = \frac{\partial r_2}{\partial u_2} \times \frac{\partial r_2}{\partial v_2} = R_2(u_2, v_2, \theta_2)\left(\frac{\partial f_2(u_2, v_2)}{\partial u_2} \times \frac{\partial f_2(u_2, v_2)}{\partial v_2}\right) \quad (5)$$

Where $r_1 = R_1(\theta_1) f_1(u_1, v_1)$ and $r_2 = R_2(u_2, v_2, \theta_2) f_2(u_2, v_2)$

Finally, there are six scalar equations out of which five can be used to find solution:

$$R_1(\theta_1)\left(\frac{\partial f_1(u_1,v_1)}{\partial u_1}\times\frac{\partial f_1(u_1,v_1)}{\partial v_1}\right)-R_2(u_2,v_2,\theta_2)\left(\frac{\partial f_2(u_2,v_2)}{\partial u_2}\times\frac{\partial f_2(u_2,v_2)}{\partial v_2}\right)=0$$

(6)

$$R_1(u_1,v_1)f_1(u_1,v_1)-a_{12}-R_2(u_2,v_2,\theta_2)f_2(u_2,v_2)=0 \quad (7)$$

Appendix B. Codes

Proposed parallel model code

```
ClearAll;
 rgn=95.80572457/2;
rgx=rgn+8.3;
B1=Pi/5;
zn=0; zx=20;
o1=Pi/2;
w1={0,0,1};w2={0,0,1};
xc1=0;yc1=0;zc1=0; center1 = {xc1,yc1,zc1};
xc2=0;yc2=106.5;zc2=0; center2 = {xc2,yc2,zc2};
a12=center2-center1;
 T=Sqrt[(u/rgn)^2-1];
a1=ArcCos[rgn/u*(Cos[T]+T*Sin[T])];
xet1=u*Cos[a1+B1*(v-zn)/(zx-zn)]+0.001u^2;
yet1=u*Sin[a1+B1*(v-zn)/(zx-zn)]+0.01u;
zet1=v;
 TransferFun1= {xet1,yet1,zet1};
 rotation1 =RotationTransform[o1,w1,{0,0,0}];
TransformedFun1= rotation1[TransferFun1]+center1;
 v2 = Dot[w2,(TransformedFun1-a12)];
u2 = Sqrt[(TransformedFun1[[1]] -a12[[1]])^2+
(TransformedFun1[[2]] -a12[[2]])^2+(TransformedFun1[[3]]
-a12[[3]])^2- v2^2];
 du21 =D[u2, u];
dv21 =D[v2, v];
 T2=Sqrt[(u2/rgn)^2-1];
a2=ArcCos[rgn/u2*(Cos[T2]+T2*Sin[T2])];
xet2=u2*Cos[a2-B1*(v2-zn)/(zx-zn)];
yet2=u2*Sin[a2-B1*(v2-zn)/(zx-zn)];
zet2=v2;
 TransferFun2={xet2,yet2,zet2};
 o2 =(2UnitStep[Dot[Cross[TransferFun2,TransformedFun1-a12],w2]]-
1)*ArcCos[Dot[TransferFun2,TransformedFun1-
a12]/Norm[TransferFun2]^2];
 rotation2 =RotationTransform[o2,w2,{0,0,0}];
TransformedFun2= rotation2[TransferFun2]+center2;
 dTu1 =D[TransferFun1,u];
dTv1 =D[TransferFun1,v];
normal1=Cross[dTu1,dTv1];
n1=rotation1[normal1];
dTu2 = D[TransferFun2,u];
```

```
dTv2 = D[TransferFun2,v];
normal2 = Cross[dTu2/du21, dTv2/dv21];
n2=rotation2[normal2];
 cross=Cross[n1,n2];
 coef1=rotation2[dTu2/du21];
coef2=rotation2[dTv2/dv21];
n=0;
results =
 Timing[FindRoot[{Dot[coef1, n1] == 0,
    Dot[coef2, n1] == 0}, {{u, rgn + 1, rgn, rgx}, {v, zn + 1, zn,
     zx}}, EvaluationMonitor :> (n++; Print[{n}])]]
{0.375,{u->51.3973,v->0.}}
 rules =results[[2]];
 o3=o2/.rules
 -1.62501
 {u3->u2/.rules,v3->v2/.rules}
 {u3->52.489,v3->0.}
 T3=Sqrt[(u3/rgn)^2-1];
a3=ArcCos[rgn/u3*(Cos[T3]+T3*Sin[T3])];
xet3=u3*Cos[a3-B1*(v3-zn)/(zx-zn)];
yet3=u3*Sin[a3-B1*(v3-zn)/(zx-zn)];
zet3=v3;
rotation3=RotationTransform[o3,w2,{0,0,0}];
TransferFun3={xet3,yet3,zet3};
TransformedFun3 = rotation3[TransferFun3]+center2;
{xa1,ya1,za1}=TransformedFun1/.rules
{xa2,ya2,za2}=TransformedFun3/.{u3->u2/.rules,v3->v2/.rules}
 {-1.43883,54.0307,0.}
 {-1.43883,54.0307,0.}
 P = Graphics3D[{PointSize[Large],Red,Point[{xa1,ya1,za1}]}];
Q = Graphics3D[{PointSize[Large],Red,Point[{xa2,ya2,za2}]}];

gear1r = ParametricPlot3D[TransformedFun1,{u,rgn,rgx},
{v,zn,zx},AxesLabel->{"x","y","z"},PlotStyle->Red];
gear2r = ParametricPlot3D[TransformedFun3,{u3,rgn,rgx},
{v3,zn,zx},AxesLabel->{"x","y","z"},PlotStyle->Blue];
                Show[gear1r,gear2r,P,Q,PlotRange-
        >Automatic,ImageSize->Full];
```

Proposed crossed model code

```
ClearAll;
 rgn=95.80572457/2;
rgx=rgn+8.3;
B1=Pi/2;
```

```
zn=0; zx=95;
o1=45°;
w1={0,0,1};w2={1,0,0};
xc1=0;yc1=0;zc1=0; center1 = {xc1,yc1,zc1};
xc2=-97.11145;yc2=105;zc2=102.981/2; center2 = {xc2,yc2,zc2};
a12=center2-center1;
 T=Sqrt[(u/rgn)^2-1];
a1=ArcCos[rgn/u*(Cos[T]+T*Sin[T])];
xet1=u*Cos[a1+B1*(v-zn)/(zx-zn)]+0.001u^2;
yet1=u*Sin[a1+B1*(v-zn)/(zx-zn)]+0.01u;
zet1=v;
 TransferFun1= {xet1,yet1,zet1};
 rotation1 =RotationTransform[o1,w1,{0,0,0}];
TransformedFun1= rotation1[TransferFun1]+center1;
 v2=Dot[w2,(TransformedFun1-a12)];
u2 =Sqrt[(TransformedFun1[[1]] -a12[[1]])^2+(TransformedFun1[[2]]
-a12[[2]])^2+(TransformedFun1[[3]] -a12[[3]])^2- v2^2];
 du21 =D[u2, u];
dv21 =D[v2, v];
 T2=Sqrt[(u2/rgn)^2-1];
a2=ArcCos[rgn/u2*(Cos[T2]+T2*Sin[T2])];
xet2=v2;
yet2=u2*Sin[a2-B1*(v2-zn)/(zx-zn)];
zet2=u2*Cos[a2-B1*(v2-zn)/(zx-zn)];
 TransferFun2={xet2,yet2,zet2};
 o2 =(2 UnitStep[Dot[Cross[TransferFun2,TransformedFun1-
a12],w2]]-1)*ArcCos[Dot[TransferFun2,TransformedFun1-
a12]/Norm[TransferFun2]^2];
 rotation2 =RotationTransform[o2,w2,{0,0,0}];
TransformedFun2= rotation2[TransferFun2]+center2;
 dTu1 =D[TransferFun1,u];
dTv1 =D[TransferFun1,v];
normal1=Cross[dTu1,dTv1];
n1=rotation1[normal1];
dTu2 = D[TransferFun2,u];
dTv2 = D[TransferFun2,v];
normal2 = Cross[dTu2/du21, dTv2/dv21];
n2=rotation2[normal2];
 cross=Cross[n1,n2];
 coef1=rotation2[dTu2/du21];
coef2=rotation2[dTv2/dv21];
n=0;
results =
 Timing[FindRoot[{Dot[coef1, n1] == 0,
    Dot[coef2, n1] == 0}, {{u, rgn + 1, rgn, rgx}, {v, zn + 1, zn,
     zx}}, EvaluationMonitor :> (n++; Print[{n}])]] rules =results[[2]];
```

```
{0.421875,{u->53.2788,v->62.146}}
 o3=o2/.rules;
 T3=Sqrt[(u3/rgn)^2-1];
a3=ArcCos[rgn/u3*(Cos[T3]+T3*Sin[T3])];
xet3=v3;
yet3=u3*Sin[a3-B1*(v3-zn)/(zx-zn)];
zet3=u3*Cos[a3-B1*(v3-zn)/(zx-zn)];
rotation3=RotationTransform[o3,w2,center2];
TransferFun3={xet3,yet3,zet3};
TransformedFun3 = rotation3[TransferFun3]+center2;

{xa1,ya1,za1}=TransformedFun1/.rules
{xa2,ya2,za2}=TransformedFun2/.rules
 {-12.8859,53.6471,62.146}
 {-12.8859,53.6471,62.146}
 P = Graphics3D[{PointSize[Large],Red,Point[{xa1,ya1,za1}]}];
Q = Graphics3D[{PointSize[Large],Red,Point[{xa2,ya2,za2}]}];

gear1r = ParametricPlot3D[TransformedFun1,{u,rgn,rgx},
{v,zn,zx},AxesLabel->{"x","y","z"},PlotStyle->Red];
gear2r = ParametricPlot3D[TransformedFun3,{u3,rgn,rgx},
{v3,95/2,3*95/2},AxesLabel->{"x","y","z"},PlotStyle->Blue];
                Show[gear1r,gear2r,P,Q,PlotRange-
        >Automatic,ImageSize->Full];
```

Commercial parallel model code

```
ClearAll;
 rgn=95.80572457/2;
rgx=rgn+8.3;
B1=Pi/5;
zn=0; zx=20;
o1=Pi/2;
w1={0,0,1};w2={0,0,1};
xc1=0;yc1=0;zc1=0; center1 = {xc1,yc1,zc1};
xc2=0;yc2=106.5;zc2=0; center2 = {xc2,yc2,zc2};
a12=center2-center1;

 T=Sqrt[(u/rgn)^2-1]; (*Anglel Theta*)
a1=ArcCos[rgn/u*(Cos[T]+T*Sin[T])]; (*Angle alpha of the initial
involute wrt x-axis*)
xet1=u*Cos[a1+B1*(v-zn)/(zx-zn)]+0.001u^2;
yet1=u*Sin[a1+B1*(v-zn)/(zx-zn)]+0.01u;
zet1=v;
```

```
 TransferFun1= {xet1,yet1,zet1};
 rotation1 =RotationTransform[o1,w1,{0,0,0}];
 TransformedFun1= rotation1[TransferFun1]+center1;
 T2=Sqrt[(u2/rgn)^2-1]; (*Anglel Theta*)
a2=ArcCos[rgn/u2*(Cos[T2]+T2*Sin[T2])]; (*Angle alpha of the initial
involute wrt x-axis*)

xet2=u2*Cos[a2-B1*(v2-zn)/(zx-zn)];
yet2=u2*Sin[a2-B1*(v2-zn)/(zx-zn)];
zet2=v2;
 TransferFun2= {xet2,yet2,zet2};
 rotation2 =RotationTransform[o2,w2,{0,0,0}];
 TransformedFun2= rotation2[TransferFun2]+center2;
 dTu1 =D[TransformedFun1,u];
dTv1 =D[TransformedFun1,v];
n1 =Cross[dTu1,dTv1];
 dTu2 = D[TransformedFun2,u2];
dTv2 = D[TransformedFun2,v2];
n2 =Cross[dTu2,dTv2];
n=0;
 results=Timing[FindRoot[{TransformedFun1==
TransformedFun2,Dot[dTu1,n2]==0,Dot[dTv1,n2]==0},
{{u,51.39733674677552`,rgn,rgx},{v,0,zn,zx},
{u2,52.48902285739788`,rgn,rgx},{v2,0,zn,zx},{o2,-
1.6250109063477696`}}, EvaluationMonitor :> (n++; Print[{n}])]]
(*results=Timing[FindRoot[{TransformedFun1ǵ
TransformedFun2,Dot[dTu2,n1]ǵ0,Dot[dTv2,n1]ǵ0},
{{u,50.741039179751574`,rgn,rgx},
{v,1.8782766116076923`,zn,zx},{u2,53.3685491910063`,rgn,rgx},
{v2,1.9561042727684617`,zn,zx},{o2,-1.6122555621225632`}}]]*)
rules = results[[2]];
{xa1,ya1,za1}=TransformedFun1/.results[[2]]
{xa2,ya2,za2}=TransformedFun2/.results[[2]]
P = Graphics3D[{PointSize[Large],Yellow,Point[{xa1,ya1,za1}]}];
Q =
Graphics3D[{PointSize[Large],Yellow,Point[{xa2,ya2,za2}]}];gear1r
= ParametricPlot3D[TransformedFun1,{u,rgn,rgx},
{v,zn,zx},AxesLabel->{"x","y","z"},PlotStyle->Red];
gear2r = ParametricPlot3D[TransformedFun2 /.rules[[5]],
{u2,rgn,rgx},{v2,zn,zx},AxesLabel->{"x","y","z"},PlotStyle->Black];
Show[gear1r,gear2r,P,Q,PlotRange->Automatic,ImageSize->Full];
 FindRoot::reged: The point {51.3973,0.,52.489,0.,-1.62501} is at the edge of the
search region {0.,20.} in coordinate 2 and the computed search direction points
outside the region.
 {0.015625,{u->51.3973,v->0.,u2->52.489,v2->0.,o2->-1.62501}}
 {-1.43883,54.0307,0.}
```

{-1.43883,54.0307,0.}

Commercial crossed model code

```
ClearAll;
 rgn=95.80572457/2;
rgx=rgn+8.3;
B1=Pi/2;
zn=0; zx=95;
o1=45°;
w1={0,0,1};w2={1,0,0};
xc1=0;yc1=0;zc1=0; center1 = {xc1,yc1,zc1};
xc2=-97.11145;yc2=105;zc2=102.981/2; center2 = {xc2,yc2,zc2};
a12=center2-center1;
 T=Sqrt[(u/rgn)^2-1]; (*Anglel Theta*)
a1=ArcCos[rgn/u*(Cos[T]+T*Sin[T])]; (*Angle alpha of the initial
involute wrt x-axis*)
xet1=u*Cos[a1+B1*(v-zn)/(zx-zn)]+0.001u^2;
yet1=u*Sin[a1+B1*(v-zn)/(zx-zn)]+0.01u;
zet1=v;
 TransferFun1= {xet1,yet1,zet1};
 rotation1 =RotationTransform[o1,w1,{0,0,0}];
 TransformedFun1= rotation1[TransferFun1]+center1;
 T2=Sqrt[(u2/rgn)^2-1]; (*Anglel Theta*)
a2=ArcCos[rgn/u2*(Cos[T2]+T2*Sin[T2])]; (*Angle alpha of the initial
involute wrt x-axis*)
 xet2=v2;
yet2=u2*Sin[a2-B1*(v2-zn)/(zx-zn)];
zet2=u2*Cos[a2-B1*(v2-zn)/(zx-zn)];
 TransferFun2= {xet2,yet2,zet2};
 rotation2 =RotationTransform[o2,w2,{0,0,0}];
 TransformedFun2= rotation2[TransferFun2]+center2;
 dTu1 =D[TransformedFun1,u];
dTv1 =D[TransformedFun1,v];
n1 =Cross[dTu1,dTv1];
 dTu2 = D[TransformedFun2,u2];
dTv2 = D[TransformedFun2,v2];
n2 =Cross[dTu2,dTv2];
n=0;
 results=Timing[FindRoot[{TransformedFun1==
TransformedFun2,Dot[dTu2,n1]==0,Dot[dTv2,n1]==0},
{{u,53.27883775577709`,rgn,rgx},
{v,62.146000676991605`,zn,zx},
{u2,52.446746992396925`,rgn,rgx},
{v2,84.22552990163756`,zn,zx},{o2,-2.850759993450203`*^-7}},
EvaluationMonitor :> (n++; Print[{n}])]]
```

```
rules = results[[2]];
{xa1,ya1,za1}=TransformedFun1/.results[[2]]
{xa2,ya2,za2}=TransformedFun2/.results[[2]]
P = Graphics3D[{PointSize[Large],Yellow,Point[{xa1,ya1,za1}]}];
Q =
Graphics3D[{PointSize[Large],Yellow,Point[{xa2,ya2,za2}]}];gear1r
= ParametricPlot3D[TransformedFun1,{u,rgn,rgx},
{v,zn,zx},AxesLabel->{"x","y","z"},PlotStyle->Red];
gear2r = ParametricPlot3D[TransformedFun2 /.rules[[5]],
{u2,rgn,rgx},{v2,95/2,3*95/2},AxesLabel->{"x","y","z"},PlotStyle-
>Black];
Show[gear1r,gear2r,P,Q,PlotRange->Automatic,ImageSize->Full];
 {0.03125,{u->53.2788,v->62.146,u2->52.4467,v2->84.2255,o2->-
5.38542*10^-7}}
 {-12.8859,53.6471,62.146}
 {-12.8859,53.6471,62.146}
```

Profile plotter

```
ClearAll;
 rgn=95.80572457/2;
rgx=rgn+8.3;
B1=Pi/6;
zn=0; zx=250;
o1=Pi/2;
w1={0,0,1};w2={0,0,1};
xc1=0;yc1=0;zc1=0; center1 = {xc1,yc1,zc1};
xc2=0;yc2=103.9230485;zc2=0; center2 = {xc2,yc2,zc2};
a12=center2-center1;
 (*$Assumptions=v≤zmax&&v≥ zmin&&u≥ rg&&u≤ rg+15&&v2≤
zmax&&v2≥ zmin&&u2≥ rg&&u2≤ rg+15;*)
 T=Sqrt[(u/rgn)^2-1];
a1=ArcCos[rgn/u*(Cos[T]+T*Sin[T])];
xet1=u*Cos[a1+B1*(v-zn)/(zx-zn)];
yet1=u*Sin[a1+B1*(v-zn)/(zx-zn)];
zet1=v;

xet2=u*Cos[a1+B1*(v-zn)/(zx-zn)]+0.0001*u^2;
yet2=u*Sin[a1+B1*(v-zn)/(zx-zn)]+0.01*u;
zet2=v;
Manipulate[Show[ParametricPlot[{xet1/.v->k,yet1/.v->k},
{u,rgn,rgx},PlotRange->{{40,57},{0,32}},PlotStyle-
>Blue],ParametricPlot[{xet2/.v->k,yet2/.v->k},
{u,rgn,rgx},PlotRange->{{40,57},{0,32}},PlotStyle->Red]],
```

```
{k,0,zx}];
```

Gear visual in crossed configuration

```
ClearAll;
 rg=95.80572457/2;(*Involute starting radius Rg*)
B1=90°;(*Beta, helix angle*)
zmin=0; zmax=103.9230485;
o1=45°;
w1={0,0,1};w2={1,0,0};
 T=Sqrt[(u/rg)^2-1]; (*Anglel Theta*)
a1=ArcCos[rg/u*(Cos[T]+T*Sin[T])]; (*Angle alpha of the initial
involute wrt x-axis*)
xet1=u*Cos[a1+B1*(v-zmin)/(zmax-zmin)];
yet1=u*Sin[a1+B1*(v-zmin)/(zmax-zmin)];
zet1=v;
 x1 = rg Cos[o];
y1= rg Sin[o];
z1= i;
x2 =- i;
y2= rg Cos[o];
z2= rg Sin[o];
 xc1=0;yc1=0;zc1=0; center1 = {xc1,yc1,zc1};
xc2=103.92304852/2;yc2=103.92304852;zc2=95.80572457/2;
center2 = {xc2,yc2,zc2};
 TransferFun1= {xet1,yet1,zet1}+center1;
TransferFunCyl = {x1,y1,z1}+center1;
TransferFunCyl2 = {x2,y2,z2}+center2;
 gear1b = ParametricPlot3D[TransferFun1 ,{u,rg,rg+8.3},
{v,zmin,zmax}];
cyl1 = ParametricPlot3D[TransferFunCyl ,{o,0,2*Pi},{i,zmin,zmax}];
 gear1 = Show[gear1b, cyl1, PlotRange->Automatic,AxesLabel->
{"x","y","z"}];
rotation1 =RotationTransform[o1,w1,center1];
 TransformedFun1= rotation1[TransferFun1];
 gear1r = ParametricPlot3D[TransformedFun1 ,{u,rg,rg+8.3},
{v,zmin,zmax},AxesLabel->{"x","y","z"},PlotStyle->Red];
g1 = Show[cyl1,gear1r,PlotRange->Automatic,AxesLabel-
>{"x","y","z"}];
 xet2=-v;
yet2=u*Sin[a1+B1*(v-zmin)/(zmax-zmin)];
zet2=u*Cos[a1+B1*(v-zmin)/(zmax-zmin)];
 o2=3Pi/4-0.1;

TransferFun2= {xet2,yet2,zet2}+center2;
```

```
gear2b = ParametricPlot3D[TransferFun2 ,{u,rg,rg+8.3},
{v,zmin,zmax},AxesLabel->{"x","y","z"}];
cyl2 = ParametricPlot3D[TransferFunCyl2 ,{o,0,2*Pi},{i,zmin,zmax}];
 gear2 = Show[gear2b, cyl2, PlotRange->Automatic,AxesLabel->
{"x","y","z"}];
rotation2 =RotationTransform[o2,w2,center2];
 TransformedFun2= rotation2[TransferFun2];
 gear2r = ParametricPlot3D[TransformedFun2 ,{u,rg,rg+8.3},
{v,zmin,zmax},AxesLabel->{"x","y","z"},PlotStyle->Blue];
g2 =Show[cyl2,gear2r,PlotRange->Automatic,AxesLabel-
>{"x","y","z"}];
 P = Graphics3D[{PointSize[Large],Red,Point[center1]}];
Q = Graphics3D[{PointSize[Large],Red,Point[center2]}];

Show[g1,g2,P,Q ,PlotRange->Automatic,ImageSize->Full];
```

Gear visual in parallel configuration

```
ClearAll;
 rg=95.80572457/2;(*Involute starting radius Rg*)
B1=Pi/6;(*Beta, helix angle*)
zmin=0; zmax=95;
o1=90°;
w1={0,0,1};w2={0,0,1};
 T=Sqrt[(u/rg)^2-1]; (*Anglel Theta*)
a1=ArcCos[rg/u*(Cos[T]+T*Sin[T])]; (*Angle alpha of the initial
involute wrt x-axis*)
xet1=u*Cos[a1+B1*(v-zmin)/(zmax-zmin)];
yet1=u*Sin[a1+B1*(v-zmin)/(zmax-zmin)];
zet1=v;
 x1 = rg Cos[o];
y1= rg Sin[o];
z1= i;
 xc1=0;yc1=0;zc1=0; center1 = {xc1,yc1,zc1};
 TransferFun1= {xet1,yet1,zet1}+center1;
TransferFunCyl = {x1,y1,z1}+center1;
 gear1b = ParametricPlot3D[TransferFun1 ,{u,rg,rg+8.3},
{v,zmin,zmax}];
cyl1 = ParametricPlot3D[TransferFunCyl ,{o,0,2*Pi},{i,zmin,zmax}];
 gear1 = Show[gear1b, cyl1, PlotRange->Automatic,AxesLabel->
{"x","y","z"}];
rotation1 =RotationTransform[o1,w1,center1];
 TransformedFun1= rotation1[TransferFun1];
gear1r = ParametricPlot3D[TransformedFun1 ,{u,rg,rg+8.3},
{v,zmin,zmax},AxesLabel->{"x","y","z"},PlotStyle->Red];
```

```
g1 = Show[cyl1,gear1r,PlotRange->Automatic,AxesLabel-
>{"x","y","z"}];
 xet2=u*Cos[a1-B1*(v-zmin)/(zmax-zmin)];
yet2=u*Sin[a1-B1*(v-zmin)/(zmax-zmin)];
zet2=v;
 o2=-1.6;
xc2=0;yc2=103.9230485;zc2=0; center2 = {xc2,yc2,zc2};
TransferFun2= {xet2,yet2,zet2}+center2;
gear2b = ParametricPlot3D[TransferFun2 ,{u,rg,rg+8.3},
{v,zmin,zmax},AxesLabel->{"x","y","z"}];
cyl2 = ParametricPlot3D[TransferFunCyl+center2 ,{o,0,2*Pi},
{i,zmin,zmax}];
 gear2 = Show[gear2b, cyl2, PlotRange->Automatic,AxesLabel->
{"x","y","z"}];
rotation2 =RotationTransform[o2,w2,center2];
 TransformedFun2= rotation2[TransferFun2];
 gear2r = ParametricPlot3D[TransformedFun2 ,{u,rg,rg+8.3},
{v,zmin,zmax},AxesLabel->{"x","y","z"},PlotStyle->Blue];
g2 =Show[cyl2,gear2r,PlotRange->Automatic,AxesLabel-
>{"x","y","z"}];
Show[g1,g2, PlotRange->Automatic,ImageSize->Full]
```

Points of contact plotter and transmission ratio calculation in parallel

```
rgn=95.80572457/2;
rgx=rgn+8.3;
B1=Pi/5;
zn=0; zx=20;
o1=Pi/2+l/180*Pi;
w1={0,0,1};w2={0,0,1};
xc1=0;yc1=0;zc1=0; center1 = {xc1,yc1,zc1};
xc2=0;yc2=106.5;zc2=0; center2 = {xc2,yc2,zc2};
a12=center2-center1;
T=Sqrt[(u/rgn)^2-1];
a1=ArcCos[rgn/u*(Cos[T]+T*Sin[T])];
xet1=u*Cos[a1+B1*(v-zn)/(zx-zn)];
yet1=u*Sin[a1+B1*(v-zn)/(zx-zn)];
zet1=v;
TransferFun1= {xet1,yet1,zet1};
rotation1 =RotationTransform[o1,w1,{0,0,0}];
TransformedFun1= rotation1[TransferFun1]+center1;
v2 = Dot[w2,(TransformedFun1-a12)];
u2 = Sqrt[(TransformedFun1[[1]] -a12[[1]])^2+
(TransformedFun1[[2]] -a12[[2]])^2+(TransformedFun1[[3]]
-a12[[3]])^2- v2^2];
```

```
du21 =D[u2, u];
dv21 =D[v2, v];
T2=Sqrt[(u2/rgn)^2-1];
a2=ArcCos[rgn/u2*(Cos[T2]+T2*Sin[T2])];
xet2=u2*Cos[a2-B1*(v2-zn)/(zx-zn)];
yet2=u2*Sin[a2-B1*(v2-zn)/(zx-zn)];
zet2=v2;
TransferFun2={xet2,yet2,zet2};
o2 =(2UnitStep[Dot[Cross[TransferFun2,TransformedFun1-
a12],w2]/Norm[TransferFun2]^2]-
1)*ArcCos[Dot[TransferFun2,TransformedFun1-
a12]/Norm[TransferFun2]^2];
rotation2 =RotationTransform[o2,w2,{0,0,0}];
TransformedFun2= rotation2[TransferFun2]+center2;
dTu1 =D[TransferFun1,u];
dTv1 =D[TransferFun1,v];
normal1=Cross[dTu1,dTv1];
n1=rotation1[normal1];
dTu2 = D[TransferFun2,u];
dTv2 = D[TransferFun2,v];
normal2 = Cross[dTu2/du21, dTv2/dv21];
n2=rotation2[normal2];
cross=Cross[n1,n2];
coef1=rotation2[dTu2/du21];
coef2=rotation2[dTv2/dv21];

points=Quiet[Table[FindRoot[{Dot[coef1,n1]==0,Dot[coef2,n1]==0}
,{{u,rgn+1,rgn,rgx},{v,zn+1,zn,zx}}],{l,-23,18}]];

 list={u,v} /.points;
 ListPlot[list,AspectRatio->1,AxesLabel->{u,v},ImageSize->Medium];
 ListLinePlot[list,AspectRatio->1,AxesLabel->{u,v},ImageSize-
>Medium];
 pointsOnTF=TransferFun1/.l->0;
pointsOnTF=pointsOnTF/.points;
 a=ListPointPlot3D[pointsOnTF,PlotStyle->Red];
 gear1r = ParametricPlot3D[TransferFun1/.l->0,{u,rgn,rgx},
{v,zn,zx},AxesLabel->{"x","y","z"},PlotStyle->Blue];
 Show[gear1r,a,PlotRange->All,ImageSize->Medium];

 o11=Pi/2;
o12=Pi/2+10/180*Pi;
deltaO1=o12-o11;
t=10;
angVel1=deltaO1/t;
```

```
i12=Table[o21=o2/.l->-23/.points[[1]];
  o22=o2/.l->-23+m/.points[[1+m]];
  angVel2=(o22-o21)/m;-angVel2/angVel1,{m,1,41}];
            ListLinePlot[Reverse[i12],DataRange-
      >{67,108},AxesLabel->{"Angle, degree","Transmission
      ratio"},ImageSize->Medium];
```

Points of contact plotter and transmission ratio calculation in crossed

```
rgn=95.80572457/2;
rgx=rgn+8.3;
B1=Pi/2;
zn=0; zx=95;
o1=Pi/4+l/180*Pi;
w1={0,0,1};w2={1,0,0};
xc1=0;yc1=0;zc1=0; center1 = {xc1,yc1,zc1};
xc2=-97.11145;yc2=105;zc2=102.981/2; center2 = {xc2,yc2,zc2};
a12=center2-center1;
T=Sqrt[(u/rgn)^2-1];
a1=ArcCos[rgn/u*(Cos[T]+T*Sin[T])];
xet1=u*Cos[a1+B1*(v-zn)/(zx-zn)];
yet1=u*Sin[a1+B1*(v-zn)/(zx-zn)];
zet1=v;
TransferFun1= {xet1,yet1,zet1};
rotation1 =RotationTransform[o1,w1,{0,0,0}];
TransformedFun1= rotation1[TransferFun1]+center1;
v2 = Dot[w2,(TransformedFun1-a12)];
u2 = Sqrt[(TransformedFun1[[1]] -a12[[1]])^2+
(TransformedFun1[[2]] -a12[[2]])^2+(TransformedFun1[[3]]
-a12[[3]])^2- v2^2];
du21 =D[u2, u];
dv21 =D[v2, v];
T2=Sqrt[(u2/rgn)^2-1];
a2=ArcCos[rgn/u2*(Cos[T2]+T2*Sin[T2])];
xet2=v2;
yet2=u2*Sin[a2-B1*(v2-zn)/(zx-zn)];
zet2=u2*Cos[a2-B1*(v2-zn)/(zx-zn)];
TransferFun2={xet2,yet2,zet2};
o2 =(2UnitStep[Dot[Cross[TransferFun2,TransformedFun1-
a12],w2]/Norm[TransferFun2]^2]-
1)*ArcCos[Dot[TransferFun2,TransformedFun1-
a12]/Norm[TransferFun2]^2];
rotation2 =RotationTransform[o2,w2,{0,0,0}];
TransformedFun2= rotation2[TransferFun2]+center2;
```

```
dTu1 =D[TransferFun1,u];
dTv1 =D[TransferFun1,v];
normal1=Cross[dTu1,dTv1];
n1=rotation1[normal1];
dTu2 = D[TransferFun2,u];
dTv2 = D[TransferFun2,v];
normal2 = Cross[dTu2/du21, dTv2/dv21];
n2=rotation2[normal2];
cross=Cross[n1,n2];
coef1=rotation2[dTu2/du21];
coef2=rotation2[dTv2/dv21];

points=Quiet[Table[FindRoot[{Dot[coef1,n1]==0,Dot[coef2,n1]==0}
,{{u,rgn+1,rgn,rgx},{v,zn+1,zn,zx}}],{l,-5,70}]];
 list={u,v}/.points;
ListPlot[list,AspectRatio->1,PlotRange->All,ImageSize-
>Medium,AxesLabel->{u,v}];
 ListLinePlot[list,AspectRatio->1,PlotRange->All,ImageSize-
>Medium,AxesLabel->{u,v}];
 rotPoints=TransferFun1/.l->0;
rotPoints=rotPoints/.points;
a=ListPointPlot3D[rotPoints,PlotStyle->Red];
gear1r = ParametricPlot3D[TransferFun1/.l->0,{u,rgn,rgx},
{v,0,95},AxesLabel->{"x","y","z"},PlotStyle->Blue];
Show[gear1r,a,PlotRange->All, ImageSize->Medium];
 o11=Pi/4;
o12=Pi/4+10/180*Pi;
deltaO1=o12-o11;
t=10;
angVel1=deltaO1/t;
 i12=Table[o21=o2/.l->-5/.points[[1]];
   o22=o2/.l->-5+m/.points[[1+m]];
   angVel2=(o22-o21)/m;angVel2/angVel1,{m,1,75}];
            ListLinePlot[Reverse[i12],AxesLabel->{"Angle,
      degree","Transmission ratio"},DataRange-
      >{39,109},ImageSize->Medium];
```

Computation time calculation of commercial model in parallel configuration

```
ClearAll;
rgn=95.80572457/2;
rgx=rgn+8.3;
B1=Pi/6;
```

```
zn=0; zx=20;
o1=Pi/2;
w1={0,0,1};w2={0,0,1};
xc1=0;yc1=0;zc1=0; center1 = {xc1,yc1,zc1};
xc2=0;yc2=105;zc2=0; center2 = {xc2,yc2,zc2};
a12=center2-center1;
 T=Sqrt[(u/rgn)^2-1]; (*Anglel Theta*)
a1=ArcCos[rgn/u*(Cos[T]+T*Sin[T])]; (*Angle alpha of the initial
involute wrt x-axis*)
xet1=u*Cos[a1+B1*(v-zn)/(zx-zn)]+0.001u^2;
yet1=u*Sin[a1+B1*(v-zn)/(zx-zn)]+0.01u;
zet1=v;
 TransferFun1= {xet1,yet1,zet1}+center1;
 rotation1 =RotationTransform[o1,w1,center1];
 TransformedFun1= rotation1[TransferFun1];
 T2=Sqrt[(u2/rgn)^2-1]; (*Anglel Theta*)
a2=ArcCos[rgn/u2*(Cos[T2]+T2*Sin[T2])]; (*Angle alpha of the initial
involute wrt x-axis*)
xet2=u2*Cos[a2-B1*(v2-zn)/(zx-zn)];
yet2=u2*Sin[a2-B1*(v2-zn)/(zx-zn)];
zet2=v2;
 TransferFun2= {xet2,yet2,zet2}+center2;
 rotation2 =RotationTransform[o2,w2,center2];
 TransformedFun2= rotation2[TransferFun2];
 dTu1 =D[TransformedFun1,u];
dTv1 =D[TransformedFun1,v];
n1 =Cross[dTu1,dTv1];
 dTu2 = D[TransformedFun2,u2];
dTv2 = D[TransformedFun2,v2];
n2 =Cross[dTu2,dTv2];
 list=Table[ Quiet[AbsoluteTiming[FindRoot[{TransformedFun1==
TransformedFun2,Dot[dTu1,n2]==0,Dot[dTv1,n2]==0},
{{u,RandomReal[{51.39733674677552-
1,51.39733674677552+1}],51.39733674677552-
1,51.39733674677552+1},{v,RandomReal[{zn,5}],zn,5},
{u2,RandomReal[{52.48902285739788-
1,52.48902285739788+1}],52.48902285739788-
1,52.48902285739788+1},{v2,RandomReal[{zn,5}],zn,5},
{o2,RandomReal[{-1.7,-1.5}], -1.7,-1.5}}]][[1]]],{n,4000}];
 estDist=EstimatedDistribution[list, GammaDistribution[a,b,c,d]]
 GammaDistribution[247.87,1.56858*10^-17,0.168436,0.0178512]
 estPdf=Plot[PDF[estDist,x],{x,0,0.03},PlotRange->Full]

                Show[Histogram[list],estPdf]
```

Computation time calculation of commercial model in crossed configuration

```
ClearAll;
 rgn=95.80572457/2;
rgx=rgn+8.3;
B1=Pi/2;
zn=0; zx=95;
o1=45°;
w1={0,0,1};w2={1,0,0};
xc1=0;yc1=0;zc1=0; center1 = {xc1,yc1,zc1};
xc2=-97.11145;yc2=105;zc2=102.981/2; center2 = {xc2,yc2,zc2};
a12=center2-center1;
 T=Sqrt[(u/rgn)^2-1]; (*Anglel Theta*)
a1=ArcCos[rgn/u*(Cos[T]+T*Sin[T])]; (*Angle alpha of the initial
involute wrt x-axis*)
xet1=u*Cos[a1+B1*(v-zn)/(zx-zn)]+0.001u^2;
yet1=u*Sin[a1+B1*(v-zn)/(zx-zn)]+0.01u;
zet1=v;
 TransferFun1= {xet1,yet1,zet1}+center1;
 rotation1 =RotationTransform[o1,w1,center1];
 TransformedFun1= rotation1[TransferFun1];
 T2=Sqrt[(u2/rgn)^2-1]; (*Anglel Theta*)
a2=ArcCos[rgn/u2*(Cos[T2]+T2*Sin[T2])]; (*Angle alpha of the initial
involute wrt x-axis*)
xet2=v2;
yet2=u2*Sin[a2-B1*(v2-zn)/(zx-zn)];
zet2=u2*Cos[a2-B1*(v2-zn)/(zx-zn)];
 TransferFun2= {xet2,yet2,zet2}+center2;
 rotation2 =RotationTransform[o2,w2,center2];
 TransformedFun2= rotation2[TransferFun2];
 dTu1 =D[TransformedFun1,u];
dTv1 =D[TransformedFun1,v];
n1 =Cross[dTu1,dTv1];
 dTu2 = D[TransformedFun2,u2];
dTv2 = D[TransformedFun2,v2];
n2 =Cross[dTu2,dTv2];
 list=Table[ Quiet[AbsoluteTiming[FindRoot[{TransformedFun1==
TransformedFun2,Dot[dTu1,n2]==0,Dot[dTv1,n2]==0},
{{u,RandomReal[{51.39733674677552-
1,51.39733674677552+1}],51.39733674677552-
1,51.39733674677552+1},{v,RandomReal[{zn,5}],zn,5},
{u2,RandomReal[{52.48902285739788-
1,52.48902285739788+1}],52.48902285739788-
```

```
1,52.48902285739788+1},{v2,RandomReal[{zn,5}],zn,5},
{o2,RandomReal[{-1.7,-1.5}], -1.7,-1.5}}][[1]]],{n,2000}];

 estDist=EstimatedDistribution[list, GammaDistribution[b,n,c,d]]
 GammaDistribution[10.2604,0.000148334,0.765572,0.0232139]
 estPdf=Plot[PDF[estDist,x],{x,0,0.035},PlotRange->Full]
 Show[Histogram[list],estPdf]
```