CASE STUDY CYBER PHYSICAL PRODUCTION SYSTEMS USING AM

Instructional tutorial for the Lua script for IceSl under the topic of

"INVOLUTE GEARING: CORRECT FILLET GEOMETRY/UNDERCUT"

GROUP 11

AKSHAY KARAYI (22302496)
VENKAT RAMANAN RAJASEKAR (22308591)
KOWSHIK KARNAM KESAVALU (22304581)

Research advisor:

—Prof. Dr. -Ing. Stefan Scherbath

IcesL and Lua

IceSL Forge is a comprehensive software that facilitates both modeling and slicing, providing flexibility for creating and fabricating complex designs. Modeling is achieved using Lua scripts, enabling parameter customization to shape various designs. This technology generates printer instructions, such as G-code, efficiently without creating a mesh. IceSL Slicer takes this a step further by generating printing commands from either an STL file or a Lua script model, using slicing technology to produce the necessary G-code for 3D printing.

Introduction

This tutorial assists in the development of Involute Gearing with correct fillet geometry and undercuts for 3D printing using Lua script in IceSl. The guide walks users through creating essential parts for assembling an Involute Gears system with customization parameters.

It aims to help users understand various parameters and replicate the results. The entire parametric model is demonstrated within the Lua platform. Instructions for extracting necessary '.stl' files and G-codes for cross-platform applications and 3D printing are also included. The demonstrated model primarily covers the modeling of two components: Gear and Pinion.

Script overview

The Script "Group11.lua" allows users to create models with user-defined parameters and prepare them for 3D printing. This .lua file can be opened and edited with editors compatible with IceSL Forge or IceSL Slicer.

This scripting structure is divided into the below mentioned parts:

- Input parameters
- Basic Functions
- Generation of tooth profile
- Calculating the center distance
- Generating the gear profile
- Gear assembly

What is undercut

An undercut occurs when material is removed from the base of the gear tooth beneath the pitch circle, typically due to the involute gear cutting process. This removal or thinning weakens the gear teeth at the root, potentially compromising the gear's strength and durability.Occurrence: Undercut happens at the root of the gear teeth, below the pitch circle. Consequences: Undercutting decreases the thickness of the gear tooth base, resulting in weaker teeth that are more susceptible to failure under load. It also disrupts the ideal load distribution across the tooth profile.

1.Input Parameters

The functional components of the involute gear are obtained from the user. The parameters can be adjusted to have a complete view on the design of the spur gear and the understanding of the undercut. The tweak box enables users to change values and examine variations in the model.

A User Interface Box in IceSL has been designed for components to handle the following parameters:

1. Number of teeth (Default value, Minimum value, Maximum Value)
2. Module (Default value, Minimum value, Maximum Value)
3. Pressure angle (Default value, Minimum value, Maximum Value)
4. Face width (Default value, Minimum value, Maximum Value)
5. Fillet Radius (Default value, Minimum value, Maximum Value)
6. Profile shift Factor (Default value, Minimum value, Maximum Value)

```
no_of_teeth=ui_number("Number Of Teeth",16,14,25);
module_gear=ui_number("Module",3,2,25);
pressure_angle=ui_scalar("Pressure Angle",20,20,25);
width=ui_scalar("Face Width",15,5,20);
rotation = ui_numberBox ("Rotation of gear",0)*2;
f_r=ui_number("Fillet radius", 2,0,3)profile_shift=ui_scalar("Profile Shift Factor",0,-0.3,1);
```

Fig.1 list of input parameters

The parameters mentioned above are configured in the user interface as shown:

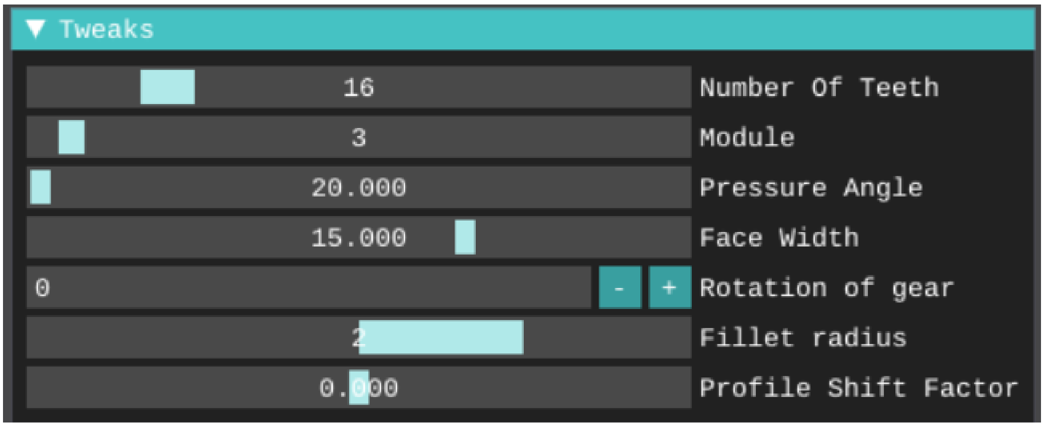

Fig.2 Tweakbox

The input parameters given by the user is initialized using the below function

```
    --- Function to obtain Input parameters

gear_parameters = function(z, m_t, alpha_t, c, x_coef, f_r, b)

    local involute_xy = {}

    m_t = m_t        -- transverse module of gear
    alpha_t = alpha_t    -- pressure angle
    z = z            -- number of teeth
    x_coef = x_coef -- profile shift coefficient
    c = 0.167 * m_t      -- clearance of tooth
    f_r =  f_r             -- fillet radius
```

Fig.3 Initialisation of input parameters

2. Deriving Gear Parameters

In this stage of the code, we derive the important parameters required for the design of the involute gear using the functional parameters obtained from the user.

```
--- Assigning Gear parameters

  d_p = z * m_t -- (1) pitch circle diameter
  r_p = d_p / 2 -- pitch circle radius

  d_b = d_p * math.cos(alpha_t)-- (2) base circle diameter
  r_b = d_b / 2 -- base circle radius

  d_a = (d_p + (2 * m_t * (1 + x_coef)))--(3) addendum circle diameter
  r_a = d_a / 2 -- addendum circle radius

  h_a = m_t*(1 +x_coef) + c -- (4) addendum height


  d_f = m_t * (z - 2) - 2 * c    -- (5) root circle diameter
  r_f = d_f / 2 -- root circle radius

  h_r = m_t + c-- (6) root height
```

Fig.4 Gear Parameters

The Involute form radius, gear tooth thickness and involute angle along the form radius are obtained using the defined input parameters.

```
 d_involute = math.sqrt(math.pow(d_p * math.sin(alpha_t) - 2 *(h_a - (m_t * x_coef) - h_r *
(1 - math.sin(alpha_t))), 2) + d_b * d_b)
--- using the equation (1),(2),(4) and (6) we obtain the (7)Involute diameter

r_involute = d_involute / 2-- form radius

---Tooth thickness calculation on the form circle

s = m_t*((math.pi/2))+ 2*x_coef* math.tan(alpha_t)--(8) thickness of the tooth along the pitch circle

angle_f = math.acos((d_p*math.cos(alpha_t))/d_involute)-- (9) involute angle along the form circle

angle_Ff = math.tan(angle_f) - angle_f;        -- (10)involute function of the involute angle

s_f = d_involute*((s/d_p)+inv_a - angle_Ff)+(x_coef*math.tan(angle_f)); --From the above equations (11) the thickness of the gear teeth obtained

omega = (s_f)/(0.5*d_involute); -- (12) angle swept along the form circle
                ----------------------
--- Function to find the slope of the given curve
slope = function(shape)
    return ((shape[2].y - shape[1].y) / (shape[2].x - shape[1].x))
end
```
Fig 5. Involute Parameters

3. Basic functions for tooth profile generation

In the next section some basic functions are defined to create the gear tooth profile

Function 1: circle function returns a vector containing the locus of a circle for the given center coordinates and the radius value.

```
circle = function(x0, y0, r, th)
return (v(x0 + r * math.cos(th), y0 + r * math.sin(th)))
end
```
Fig.6 Circle Function

Function 2: The **involute curve** function will return a vector with locus of an involute using the generating circle radius and angle of involute.

```
--- Function to deliver the x and y coordinates of the involute function in relation to the base radius and involute angle
involutecurve = function(base_r, involuteangle)
return v(base_r *(math.sin(involuteangle) - involuteangle *math.cos(involuteangle)), base_r *
(math.cos(involuteangle) + involuteangle *
math.sin(involuteangle)))
end
```
Fig.7 Involute Function

Function 3: Mirror function returns the reflection of a contour along the x-axis. This function is used to reflect the obtained involute and the root fillet curves to obtain a complete gear tooth profile.

```
--- Function to mirror the Involute curve w.r.t Y-axis
Mirror = function(xy_value)
return v(-xy_value.x, xy_value.y)
end
```
Fig.8 Mirror Function

Function 4: Rotation function is defined to rotate a curve by the given angle. This is used reciprocate multiple teeth after generating the first gear tooth profile.

```
--- Function to rotate the individual tooth profile relative to the circle center.
Rotation = function(angle, xy_value)
return v(math.cos(angle) * xy_value.x + math.sin(angle) * xy_value.y, math.cos(angle) * xy_value.y - math.sin(angle) * xy_value.x)
end
```

Fig.9 Rotation Function

Function 5: Angle function returns the slope of a given line used to obtain the involute angle between addendum circle and base circle.

```
Angle = function(r2, r1)
return (math.sqrt((r1 * r1 - r2 * r2) / (r2 * r2)))
end
```

Fig.10 Angle function

Function 6: Slope function returns the slope of a given curve. The function takes entire curve and finds the slope of the curve , not the individual points.

```
--- Function to find the slope of the given curve
slope = function(shape)
    return ((shape[2].y - shape[1].y) / (shape[2].x - shape[1].x))
end
```

Fig.11 Slope Function

4. Generating the gear tooth profile

The fillet curve is initial generated then the following profile is extended to form the involute. To start generating the fillet we obtain the center of the fillet trochoid. The start angle and stop angle is specified between base circle and the addendum circle.

```
    start_angle = Angle(r_b, r_b)
    stop_angle = Angle(r_b, r_a)
    resl = 100

-- Intiating of fillet
        local points = {}
    for i = 1, resl do
        points[i] = involutecurve(r_b, (start_angle + (stop_angle - start_angle) * i / resl))
    end

    m_s = slope(points)
    slope_angle = math.atan(m_s)
    parallel = {}-- parallel line point

    parallel[1] = v(points[1].x + f_r * math.cos(slope_angle + math.pi / 2),points[1].y + f_r * math.sin(slope_angle + math.pi / 2))
 d = (parallel[1].y - m_s * parallel[1].x) / math.sqrt(m_s * m_s + 1)
    angle1 = math.asin(d / (f_r + r_b)) + slope_angle
    center_f = {} --Finding the center of the fillet curve
    center_f[1] = v((f_r + r_f) * math.cos(angle1), (f_r + r_f) * math.sin(angle1))
```

Fig.12 Finding the Center of the Fillet Curve

The fillet curve is generated using the center coordintes obtained in the previous step. The generated locus of the fillet is stored in the involute_xy vector

```
--- Generating the fillet curve

    slope_c = 2 * math.pi + math.atan(center_f[1].y / center_f[1].x)
    slope_a = 3 * math.pi / 2 + slope_angle

    for m = 1, z do
        th = 2* math.pi

        for i = 1, resl do
            involute_xy[#involute_xy + 1] =Rotation(th*m/z,circle(center_f[1].x,
                                                            center_f[1].y, f_r,
                                                            (slope_c + (slope_a - slope_c) * i / resl)))  -- fillet radius
    end
```

Fig.13 Generation of Fillet

The fillet is generated and the locus is stored in the vector involute xy where #involute_xyth element of the vector specifies the intersection point of the involute and the fillet profile

The involute locus coordinates starts from #involute_xy+1 where involute curve function is used to generate involute curve taken stop angle between base radius and involute radius and the start angle between base circle radius and addendum radius.

```
--- Generating tooth profile

    for i = 1, resl do
        involute_xy[#involute_xy + 1] =Rotation(th*m/z, involutecurve(r_b, (start_angle +(stop_angle -start_angle) *i / resl)))-- actual involute
      end
```

Fig.14 Generation of the Involute

The generated locus of involute and fillet profile is mirrored using the mirror function to develop the other side of the tooth profile. The generated profile is created as a 2D curve using predefined linear_extrude function.

```
        for i = resl, 1, -1 do
          involute_xy[#involute_xy + 1] =Rotation(th*m/z,Rotation(tooth_ang, Mirror(involutecurve(r_b,(start_angle +(stop_angle -start_angle) *i / resl)))))

         end
    end


    involute_xy[#involute_xy + 1] = involute_xy[1]
    output = linear_extrude(v(0,0,width),involute_xy) -- actual tooth profile is developed
    return output
end
```

Fig.15 Mirroring the Tooth Profile

Extrude function is defined to create the cylindrical bore generated at the center of the gear

```
function extrude(Contour, angle, dirctn_vector, scale_vector, z_steps)
    local no_contour= #Contour -- Number of points in the contour
    local Vertex= {}

---Generating the contour
   for
      j= 0,z_steps-1 do
      local phi= angle*j/(z_steps-1)
      local dirctn_vectorh= dirctn_vector*j/(z_steps-1)
      local scale_vectorh= (scale_vector - v(1,1,1))*(j/(z_steps-1)) + v(1,1,1)

      for
          i= 1,no_contour-1 do
          Vertex[i+j*no_contour]= v((dirctn_vectorh.x + scale_vectorh.x * (Contour[i].x*math.cos(phi) - Contour[i].y*math.sin(phi))),
 (dirctn_vectorh.y + scale_vectorh.y * (Contour[i].x*math.sin(phi) + Contour[i].y*math.cos(phi))),
                              (dirctn_vectorh.z * scale_vectorh.z))
      end
```

Fig.16 Extrude Function

5. Calculating the center distance of the gear pair

The gear parameters are initialized and the number of teeth on the gear is set to have 5 more than the pinion. This adjustment is done to show the effects of undercut on pinion.

```
---intializing parameter gear and pinon
z_2 = no_of_teeth; ---pinion
z_1 = z_2+5;       ---gear
x=profile_shift
alpha_t = 20*math.pi/180  ---transverse pressure angle taken standard value
inv_a = math.tan(alpha_t) - alpha_t;---involute function
```

Fig.17 Gear Parameters Initialisation

Center distance calculation is carried out using working pressure angle. The involute function of the working pressure angle is obtained using number of teeth and profile shift coefficient. We use an approximation formula to obtain the working pressure angle from its involute function. Formula referred from Dudley's Handbook of gears.

```
-- involute function of Working pressure angle   ///
   inv_alpha_w= 2*((x+x)/(z_1+z_2))*math.tan(alpha_t) + inv_a;   --  Involute Of working pressure angle
--- calculation of working pressure angle  -----
   alpha_w=(((math.pow(3,(1/3)))*(math.pow(inv_alpha_w,(1/3)))) - (2*inv_alpha_w/5) + (math.pow(9/175*3,(2/3)))*(math.pow(inv_alpha_w,(5/3))) - (math.pow(2/175*3, (1/3))
                    *(math.pow(inv_alpha_w,(7/3))) - ((144/67375)*(math.pow(inv_alpha_w,(9/3))) + (3258/3128215)*(math.pow(3,(2/3)))*(math.pow(inv_alpha_w,(11/3)))
                    - (49711/153278125)*(math.pow(3,(1/3)))*(math.pow(inv_alpha_w,(13/3))) - (1130112/9306171875)*(math.pow(inv_alpha_w,(15/3)))
                    + (5169659643/95304506171875)*(math.pow(3,(2/3)))*(math.pow(inv_alpha_w,(17/3))))) -- approximation formula to obtain qorking pressure angle from
      --------------------

   center_d= (module_gear*(z_1+z_2)/2)*(math.cos(alpha_t)/math.cos(alpha_w))   -- Center distance: Gears using working pressure angle


addOn_distance= module_gear * 2;
```

Fig.18 Finding the Center Distance

6.Gear Generation

The pinion and gear parameters are initialized separately using 2 functions. The function call is done instantaneously so that in next step these elements can be extruded

```
function pinion() -- function to generate pinion
    ext_gr = gear({z=no_of_teeth;m_t=module_gear;alpha_t=pressure_angle*math.pi/180;c=clearance;width=width;x_coef=profile_shift;f_r=f_r})
end

pinion()  -- Calling of the function

function gear1() --Function to generate gear shape
    ext_gr2 = gear({z=no_of_teeth+5;m_t=module_gear;alpha_t=pressure_angle*math.pi/180;c=clearance;width=width;x_coef=profile_shift;f_r=f_r})
end

gear1()
angle = (((math.pi * module_gear / 2) + 2 * module_gear * profile_shift * math.tan(alpha_t)) / r_p + 2 * math.tan(alpha_t) - 2 * alpha_t)
```

Fig.19 Gear Generation Function

The initial orientation of the pinion is specified in rotation3, rotation4 and that of the gear is specified in rotation42. The gear orientations need to be adjusted for pinions with odd number of teeth hence added extra rotation of $180/z\_1$.
r2 and r3 defines the gear rotation. r3 is given as negative to specify the rotation of gear and pinon in opposite direction

```
rotation3 = rotate(0,0,angle*90/math.pi)
rotation4 = rotate(0,0,(angle*90/math.pi))
if (z_2%2==0)
then
rotation42 = rotate(0,0, ((angle*90/math.pi)+z_1/z_2))
else
rotation42 = rotate(0,0, ((angle*90/math.pi))+180/z_1+1)
end

 --- Formation of gear and pinion
r2 = rotate(0,0,rotation)
r3 = rotate(0,0,-rotation*z_2/z_1)
```

Fig.20 Orientation of the Gear and Pinion

7.Generating the Gear pair

Cylindrical shape of the bore is defined before creating the gears using the extrusion function specifying its position, length and orientation vector which is later used for material removal at the center of the gear blank

```
bore_pinion=extrude(circle_t(addOn_distance),0,v(0,0,width), v(1,1,1),20)   -- Bore formation of pinion
bore_diff=bore_pinion
bore_full=difference(ext_gr,bore_diff)

bore_gear=extrude(circle_t(addOn_distance*1.5),0,v(0,0,width), v(1,1,1),20) -- Bore formation of external gear
bore_diff2=bore_gear
bore_full2=difference(ext_gr2,bore_diff2)
```

Fig.21 Bore Generation

The gear and pinion are generated using the emit function specifying the position of the pair inside the translate function, the rotation is specified by r2 and r3 and the initial orientation is given by rotation4 and rotation42. The difference function removes a cylindrical bore In the middle of the gear blank.

```
emit(translate(0,0,40)*r2*rotation4*difference(bore_full,bore_diff),5)
    -- creates an pinion
emit(translate(0,center_d,40)*r3*rotation42*difference(bore_full2,bore_diff2),7)
    -- creates an gear
```

Fig.22 Gear and Pinion generation.

Assembly Description:

Part List:

1. Gear
2. Pinion
3. Shaft

The Gear and the Pinion can be printed by the user by the above given instructions and Parameters and inputs. Further Assembly parts are added to support of the gear pair.

Shaft

2 cylinders are created to act as shaft for gear and pinion using the predefined function cylinder. The position of the shaft exactly matches that of the gear pair and the radius of the shaft is same as the radius of the bore present in the center of the gear blank.

```
-- Define the radius and height of the shaft
local radius = addOn_distance
local height = 80

-- Create a cylinder representing the shaft
shaft = cylinder(radius, height)

-- Set the shaft's position (optional, center it along the z-axis)
shaft = translate(0, 0,0) * shaft

-- Render the shaft
emit(shaft) -- Shaft 1
shaft2 = cylinder(radius*1.5, height) --Shaft 2
shaft2 = translate(0, center_d,0) * shaft2
emit(shaft2)
```

Fig.23 Shaft Generation

Base Plate

A cuboid is added for the base plate to hold the pair using the predefined function cube.

```
cube=ccube(180,200,1) -- Creation of base plate
cube = translate(4, 20,0) * cube
emit(cube)
```

Fig.24 Base Plate generation

Gear Tooth Assembly:

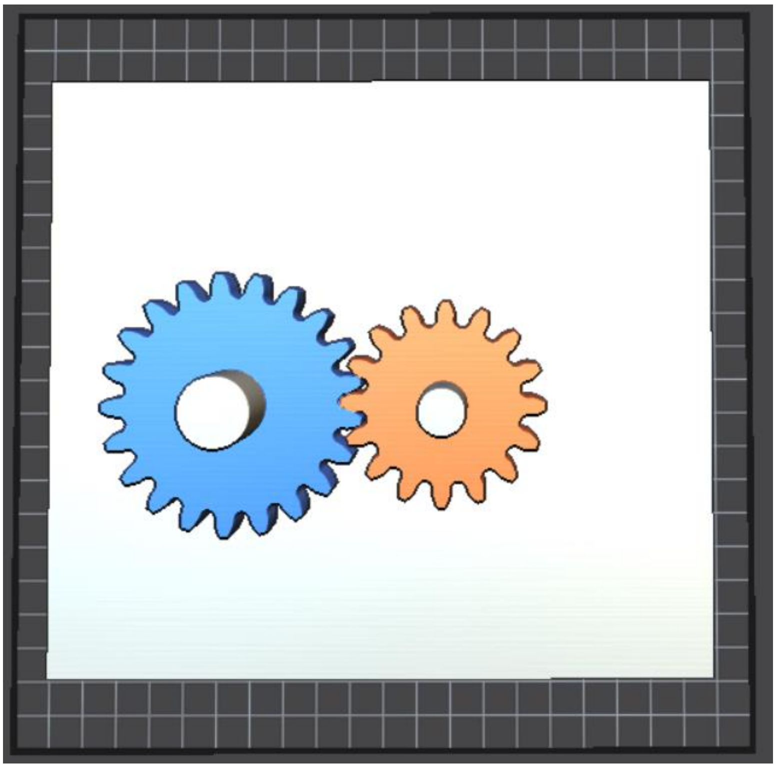

Fig.25 assembly top view

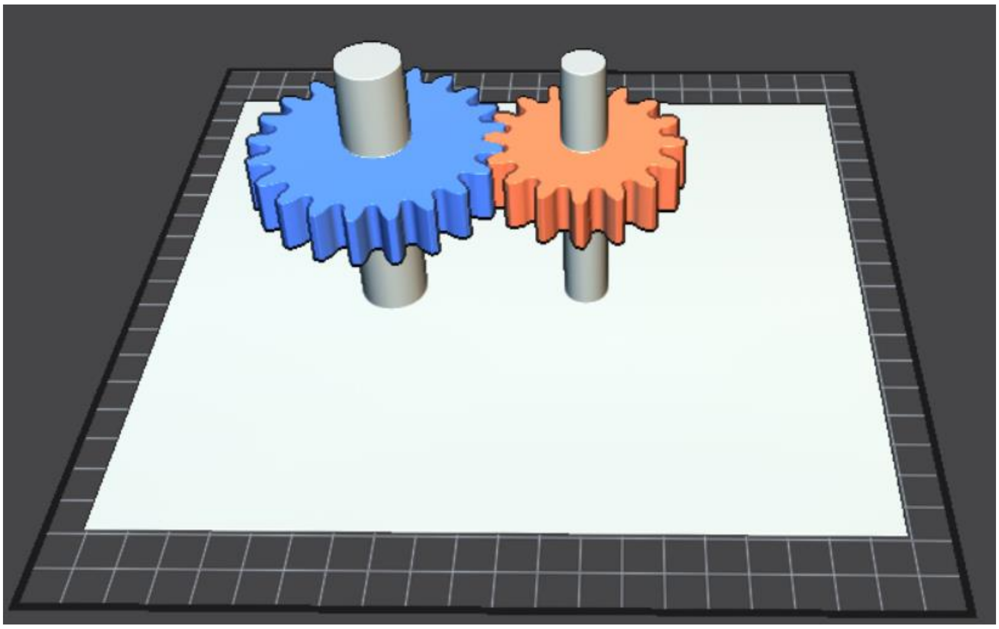

Fig .26 Meshing

Reference:

[1]: Zhai, Guodong & Liang, Zhihao & Fu, Zihao. (2020). A Mathematical Model for Parametric Tooth Profile of Spur Gears. Mathematical Problems in Engineering. 2020.

[2]: HANDBOOK OF DUDLEY'S. Practical gear design and manufacture.

[3]: J I Pedrero, M Artes, C Garcia-Masia (2004). Determination of the effective path of contact of undercut involute gear teeth.Institution of Mechanical Engineers Part C Journal of Mechanical Engineering Science 1989-1996 (vols 203-210) 218(7):751-760