МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра “Металлорежущие станки и инструменты”

ОТЧЕТ по лабораторной работе №26

**Автоматизация моделирования косозубой эвольвентной передачи в среде САПР высшего уровня**

Выполнил:

студент группы ИСиТ-221

Мельников А.В.

Проверил:

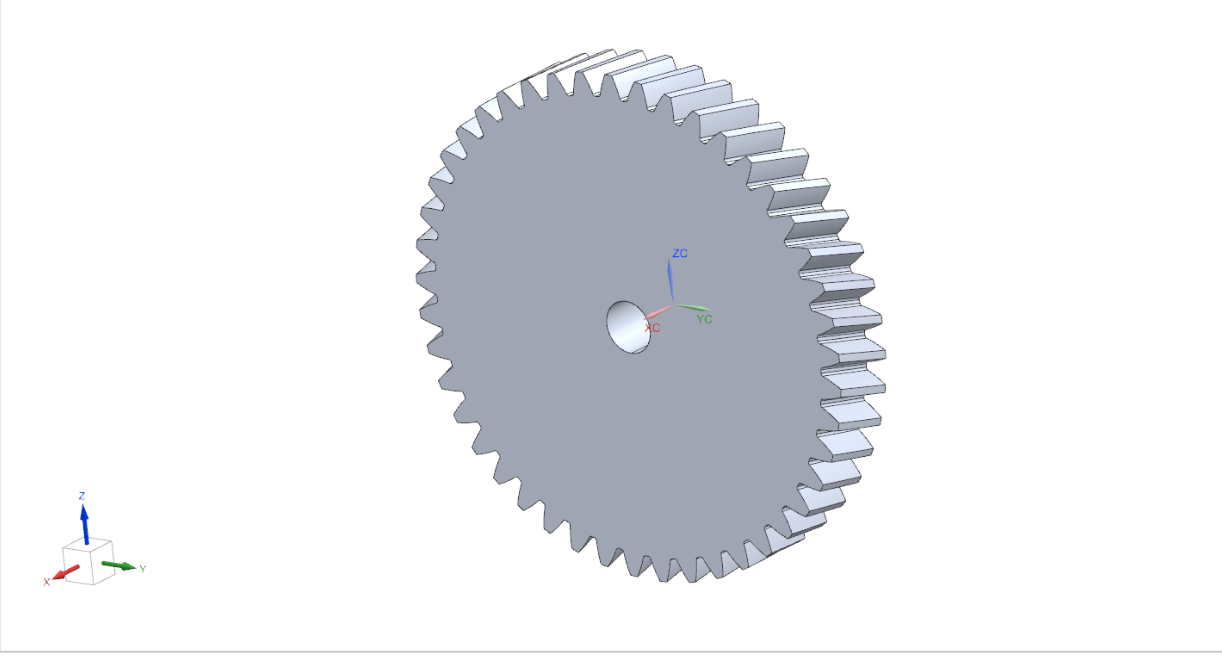
Доцент

Хатетовский С.Н.

Могилёв, 2024

***Цель работы***:изучение основных принципов автоматизации моделирования косозубой эвольвентной передачи в среде NX.

***Перечень используемого оборудования и программного обеспечения***:персональный компьютер, САПР NX.



using System;

using NXOpen;

using NXOpen.UF;

public class GearCreator

{

public static void Main()

{

Session theSession = Session.GetSession();

Part workPart = theSession.Parts.Work;

Part displayPart = theSession.Parts.Display;

int numberOfTeeth = 45;

double width = 20.0;

double module = 3.0;

double centerHoleDiameter = 15.0;

double pitchDiameter = numberOfTeeth \* module;

double outerDiameter = pitchDiameter + 2 \* module;

double innerDiameter = centerHoleDiameter;

double toothThickness = module \* 1.5;

double toothHeight = module \* 1.5;

double angleOffset = 15.0;

try

{

for (int i = 0; i < numberOfTeeth; i++)

{

double angle = (360.0 / numberOfTeeth) \* i;

double x1 = (outerDiameter / 2.0) \* Math.Cos(DegreesToRadians(angle));

double y1 = (outerDiameter / 2.0) \* Math.Sin(DegreesToRadians(angle));

double x2 = ((outerDiameter / 2.0) - toothThickness) \* Math.Cos(DegreesToRadians(angle + (360.0 / numberOfTeeth) / 2));

double y2 = ((outerDiameter / 2.0) - toothThickness) \* Math.Sin(DegreesToRadians(angle + (360.0 / numberOfTeeth) / 2));

double x3 = (outerDiameter / 2.0) \* Math.Cos(DegreesToRadians(angle + (360.0 / numberOfTeeth)));

double y3 = (outerDiameter / 2.0) \* Math.Sin(DegreesToRadians(angle + (360.0 / numberOfTeeth)));

double zOffset = width \* Math.Tan(DegreesToRadians(angleOffset));

NXOpen.Line line1 = workPart.CurveCollection.CreateLine(new Point3d(x1, y1, 0), new Point3d(x2, y2, zOffset), null);

NXOpen.Line line2 = workPart.CurveCollection.CreateLine(new Point3d(x2, y2, zOffset), new Point3d(x3, y3, 0), null);

NXOpen.Line line3 = workPart.CurveCollection.CreateLine(new Point3d(x3, y3, 0), new Point3d(x1, y1, 0), null);

}

Cylinder cylinder = workPart.Cylinder.CreateCylinder(new Point3d(0, 0, 0), outerDiameter / 2.0, width, null);

cylinder.Commit();

Hole hole = workPart.Hole.CreateHole(new Point3d(0, 0, 0), centerHoleDiameter / 2.0, width, null);

hole.Commit();

}

catch (Exception ex)

{

Console.WriteLine($"Ошибка: {ex.Message}");

}

workPart.Update();

}

private static double DegreesToRadians(double degrees)

{

return degrees \* (Math.PI / 180.0);

}

}