****

****

**题 目 TY－460 自动平台轮转印刷机设计计算说明书**

**课程名称 机械原理课程设计**

**姓 名 徐屹寒**

**学 号 3230103743**

**指导老师 李立新**

**专业班级 机械2305**

**年 级 2023级**

# 1.滚筒传动链的参数设计和运动分析

## 1.1给定GN与AD之间的水平间距w＝290mm，根据第五项中的参考尺寸，设计变位齿轮 1、2。

两齿轮实际中心距：

已知,故

，

取，并计算得理论中心距

取，利用中心距和啮合角函数方程及无侧隙啮合方程：

其中，我们取

已知，计算变位齿轮的其他参数：

，

，

## 1.2参考第五项中的尺寸，用作图法求出双曲柄机构GHMN的12个位置图以及与之相应的速度矢量图，据此确定相应位置的滚筒圆周速度（每隔 30一个），并标明其中双曲柄机构GHMN的最小传动角所在位置。

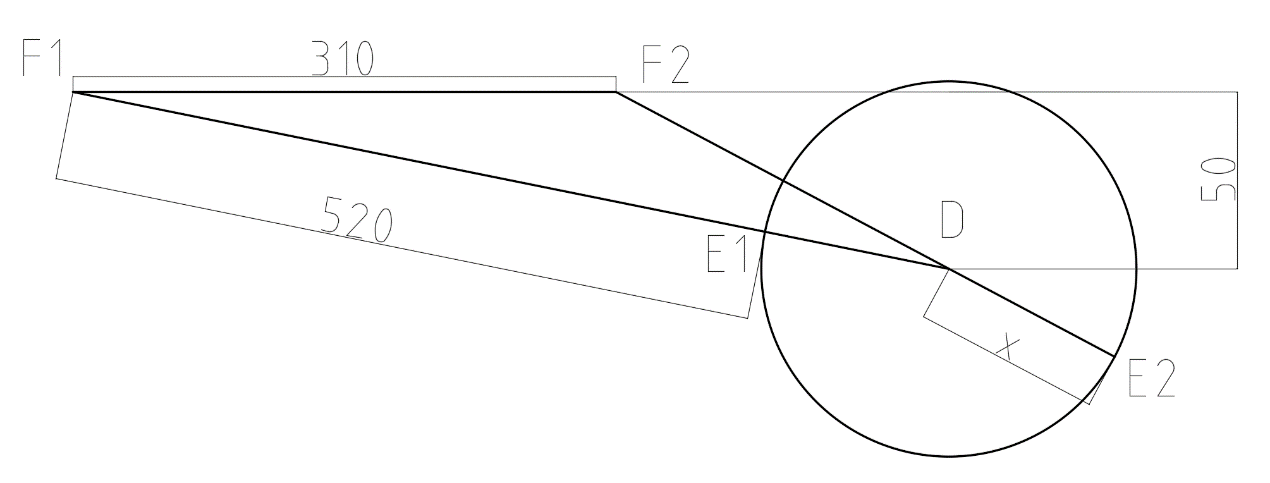
见附图一

# 2.平台传动链的机构选型、参数设计和运动分析

## 2.1给定平台的行程为620mm，根据第五项中的参考尺寸，计算确定曲柄连杆机构DEF中的曲柄长度。

平台的行程为620mm，故F点的行程为310mm

, ,设



由图可知

解得

## 2.2参考第五项中的尺寸，用作图法求出双曲柄机构ABCD以及曲柄连杆机构DEF的12个位置图以及F点的12个速度矢量图（每隔30一个），并标明双曲柄机构ABCD的最小传动角所在位置。

由及初始情形下F,E,D三点共线，可以求得初始角。

具体作图见附图二

# 3.同步补偿凸轮机构设计

## 3.1采用附录中的参考数据，绘制滚筒圆周位移曲线与平台位移曲线图；绘制滚筒圆周速度曲线以及平台速度曲线图。

## 3.2确定同步区。同步区内滚筒圆周的线位移应大于印刷幅面的宽度320mm，建议取 420mm左右。同步区的起点应取在滚筒圆周速度与平台速度相等的点上，以免在补偿起点产生刚性冲击。

滚筒圆周速度与平台速度相等。

同步区的终点：由，取AB转角155。

所以同步区AB转角34-155，相对应的凸轮转角（即CD转角）范围是32.319-117.517

## 3.3分析滚筒圆周位移曲线和平台位移曲线，绘制平台在同步区的位移补偿量曲线图；设计并绘制平台在复位区的位移补偿量曲线。建议复位区凸轮转角要取得大一些，到100左右，以减小凸轮机构的回程压力角。设计复位区的位移补偿量曲线时要避免在其两端产生刚性冲击。

同步区：

从AB转角34开始补偿，补偿量计算公式为：

其中p为平台位移，q为滚筒线位移。 为平台初始位移，为滚筒初始线位移。

故同步区的位移补偿量为：

复位区：

为了减小凸轮机构的回程压力角，将复位区凸轮转角（即CD转角）取为100左右，故CD转角在220左右完成复位。

因为同步区结束后齿条6的速度不为零，为了避免产生冲击，其位移变化应先采用一段圆弧过渡，使之速度减小到0；

设从动件在凸轮转角为125°时减小到0。为保证从动件速度不发生突变，取过渡圆弧斜率k在凸轮转角为117.517°为

其中，分别为凸轮转角为117.517°时的平台速度与滚筒速度。为凸轮每秒旋转角度（弧度制）。由此可以画出进入复位区的补偿曲线,并计算得到从动件在125°与117.517°时的位移之差

同时，为了避免复位区结束时产生刚性冲击，选择摆线运动（即正弦加速度运动）使齿条6复位，同时速度减小到0。

从动件升程

从动件的摆线运动方程为，其中

复位区补偿计算matlab代码实现如下：

*%% 进入复位区圆弧参数计算*

*% 定义复位区的角度范围（度）*

Fai\_CD = [117.517;125];  *% 复位区起始角度和结束角度*

*% 定义初始位移参数*

x1 = 14.147;        *% 复位区起始位移（mm）*

v1 = 1269.937;      *% 复位区起始速度（mm/s）*

v2 = 1175.302;      *% 复位区结束速度（mm/s）*

*% 凸轮转速计算*

omiga = 4500/3600\*360;  *% 凸轮角速度：4500转/分钟转换为度/秒*

*% 计算速度变化率和倾斜角*

k = (v1-v2)/omiga;      *% 速度变化率（mm/度）*

theta\_k = atan(k);      *% 速度变化对应的角度（弧度）*

*% 计算圆弧半径和位移增量*

R = (Fai\_CD(2)-Fai\_CD(1))/sin(theta\_k);  *% 圆弧半径（mm）*

delta\_x = R\*(1-cos(theta\_k));             *% 圆弧段的位移增量（mm）*

*%% 复位区结束补偿量计算*

*% 计算复位区结束时的总位移*

h = x1+delta\_x;         *% 复位区结束位移 = 起始位移 + 圆弧位移增量*

*% 定义补偿函数参数*

Delta\_t = 100;          *% 补偿区总角度范围（度）*

*% 定义位移补偿函数*

f = @(Delta\_1) (h\*(1-(Delta\_1/Delta\_t)+1/2/pi\*sin(Delta\_1/Delta\_t\*2\*pi)));

*% 生成补偿区角度序列和对应的位移*

Delta\_1 = 0:5:100;      *% 补偿区角度范围，步长为5度*

s = f(Delta\_1);         *% 计算各角度对应的补偿位移*

*%% 绘制补偿量曲线*

*% 定义圆弧段的几何参数*

startPoint = [Fai\_CD(1), x1];    *% 圆弧起点坐标（角度，位移）*

endPoint = [Fai\_CD(2), h];       *% 圆弧终点坐标（角度，位移）*

center = [Fai\_CD(2), h-R];       *% 圆弧圆心坐标*

*% 计算起点和终点相对于圆心的角度*

theta1 = atan2(startPoint(2) - center(2), startPoint(1) - center(1));  *% 起点角度*

theta2 = atan2(endPoint(2) - center(2), endPoint(1) - center(1));      *% 终点角度*

*% 生成圆弧上的采样点*

t = linspace(theta1, theta2, 20);   *% 在起点和终点之间生成20个等间距角度*

arcX = center(1) + R \* cos(t);      *% 圆弧上各点的横坐标（角度）*

arcY = center(2) + R \* sin(t);      *% 圆弧上各点的纵坐标（位移）*

*% 组合完整的补偿曲线数据*

x = [arcX,Delta\_1+arcX(20)];        *% 横坐标：圆弧段 + 补偿段（相对于圆弧终点）*

y = [arcY,s];                       *% 纵坐标：圆弧位移 + 补偿位移*

*% 绘制最终的位移补偿曲线*

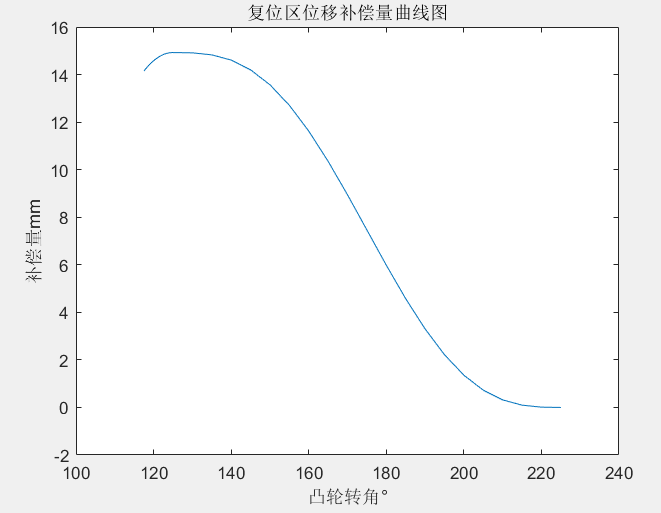
plot(x,y)

xlabel("凸轮转角°")                 *% 横轴标签*

ylabel("补偿量mm")                  *% 纵轴标签*

title("复位区位移补偿量曲线图")      *% 图标题*

最终绘制结果为：



## 3.4以凸轮转角为横坐标，绘制全工作循环内从动件的位移曲线图。

## 3.5设计主凸轮（右推凸轮10）的轮廓。

结合从动件位移曲线图，运用反转法绘出主凸轮轮廓。

## 3.6设计回凸轮（左推凸轮11）的轮廓，要求与主凸轮有相同的最大径向尺寸，并确定 。

滚子与滚子的中心距保持不变，且二者与相应凸轮的接触点保持180不变，可据此设计出回凸轮轮廓。

凸轮轮廓见附图三

凸轮的最大径向尺寸为：

最小径向尺寸为：

故。

# 4.绘制机构主运动简图

见附图四

# 5.编程计算

四杆机构求解matlab程序

*%% 设置参数*

*% 双曲柄机构ABCD的杆长参数（单位：mm）*

LAB = 116;  *% 曲柄AB的长度*

LBC = 100;  *% 连杆BC的长度*

LCD = 100;  *% 摇杆CD的长度*

LDA = 37;   *% 机架DA的长度*

*% 第二个机构GHMN的杆长参数（单位：mm）*

LGH = 100;  *% 杆GH的长度*

LHM = 100;  *% 杆HM的长度*

LMN = 95;   *% 杆MN的长度*

LNG = 40;   *% 杆NG的长度*

*% 其他机构参数*

m = 4;      *% 质量参数*

r3 = 136;   *% 半径参数3*

r4 = 136;   *% 半径参数4*

r8 = 100;   *% 半径参数8*

e = 50;     *% 偏心距（单位：mm）*

LEF = 520;  *% 杆EF的长度（单位：mm）*

a\_CDE = 177/360\*2\*pi;  *% 角度CDE（弧度制）*

*% 基圆和滚子半径参数*

rb = 90;    *% 基圆半径（单位：mm）*

rg = 25;    *% 滚子半径（单位：mm）*

*% 运动参数*

omiga = 4500/3600\*2\*pi;  *% 曲柄角速度：4500转/分钟转换为弧度/秒*

*% 初始角度设置（弧度制）*

Fai\_GH0 = 57.109/360\*2\*pi;   *% 杆GH的初始角度*

Fai\_AB0 = 115.66/360\*2\*pi;   *% 曲柄AB的初始角度*

*% 主循环：计算360度范围内的机构运动*

*for* i = 1:360

    Fai = i-1;                    *% 当前计算角度（度）*

    Fai\_AB = Fai/360\*2\*pi;        *% 将角度转换为弧度制*

*%% 双曲柄ABCD求解*

*% 调用曲柄摇杆机构求解函数，计算CD杆的角位移和角速度*

    [theta,omega] = crank\_rocker(Fai\_AB+Fai\_AB0,omiga,LAB,LBC,LCD,LDA);

*% 存储计算结果*

    theta\_CD\_list(i,2) = theta/pi\*180;  *% 将弧度转换为度并存储*

    theta\_CD\_list(i,1) = Fai;           *% 存储对应的输入角度*

    omega\_CD = omega(2);                *% CD杆的角速度*

*%% 曲柄滑块DEF求解*

*% [s,v] = slider\_crank()  % 待完善的滑块机构求解*

*end*

*% 对角位移数据进行相对化处理，以初始位置为零点*

theta\_CD\_list(:,2) = theta\_CD\_list(:,2)-theta\_CD\_list(1,2);

*% 曲柄摇杆机构求解函数*

function [theta3,omega] = crank\_rocker(theta1,omega1,l1,l2,l3,l4)

*% 输入参数：*

*% theta1 - 曲柄转角（弧度）*

*% omega1 - 曲柄角速度（弧度/秒）*

*% l1,l2,l3,l4 - 各杆长度*

*% 输出参数：*

*% theta3 - 摇杆角位移*

*% omega - 各杆角速度向量*

*%计算角位移*

    L = sqrt(l4\*l4+l1\*l1-2\*l1\*l4\*cos(theta1));

    phi = asin((l1./L)\*sin(theta1)); *%phi 记录直线BD到AD的角*

*if* l1>L  *%l1为最长边时phi为钝角*

        phi = pi/2-phi;

*end*

    beta = acos((-l2\*l2+l3\*l3+L\*L)/(2\*l3\*L)); *%beta 记录直线CD到BD的角*

*if* theta1 > pi

        phi = -phi;

*end*

    theta3 = pi - phi -beta;

    theta2 = asin((l3\*sin(theta3)-l1\*sin(theta1))/l2);

*%计算角速度*

    A  = [-l2\*sin(theta2),l3\*sin(theta3);

            l2\*cos(theta2),-l3\*cos(theta3)];

    B = [l1\*sin(theta1);-l1\*cos(theta1)];

    omega = A\(omega1\*B);

end

*% 曲柄滑块机构求解函数*

function [s3,v3] = slider\_crank(theta1,omega1,l1,l2,e)

*% 输入参数：*

*% theta1 - 曲柄转角（弧度）*

*% omega1 - 曲柄角速度（弧度/秒）*

*% l1,l2 - 曲柄和连杆长度*

*% e - 偏心距*

*% 输出参数：*

*% s3 - 滑块位移*

*% v3 - 滑块速度*

*% 计算线位移*

    theta2 = asin((e-l1\*sin(theta1))/l2);    *% 连杆与水平线夹角*

    s3 = l1\*cos(theta1)\*l2\*cos(theta2);      *% 滑块位移计算*

*% 计算线速度*

*% 建立速度方程组矩阵*

    A = [l2\*sin(theta2),1;         *% 速度约束矩阵*

         -l2\*cos(theta2),0];

    B = [-l1\*sin(theta1);          *% 已知项向量*

         l1\*cos(theta1)];

    omega = A\(omega1\*B);          *% 求解角速度和线速度*

    v3 = omega(2);                 *% 提取滑块线速度*

end