

# Курсовая: Техническая механика

## Расчет червячной передачи

```
In [1]: from math import pi, atan, degrees, cos, tan, radians

def degrees_to_radians(degrees, minutes=0):
    total_degrees = degrees + minutes / 60

    # Переводим градусы в радианы
    return radians(total_degrees)
```

```
In [2]: # Частота вращения выходного вала редуктора
n_out = 59
```

## Выбор ЭД

```
In [3]: # КПД общий
# КПД передачи = 0.8 (таблица 1.1)
kpd_m = 0.98
kpd_m * 0.8
```

Out[3]: 0.784

```
In [4]: # Мощность ЭД
0.9 / 0.784
```

Out[4]: 1.1479591836734693

```
In [5]: # Диапазон частот
print(f"{59*15} ... {59*30}")
```

885 ... 1770

---

Выбираю ЭД

4A80B4Y3 (3 - Чернавский стр.26)

- Частота оборотов: 1500 Об/мин
- Мощность: 1500 Вт
- Скольжение: 5.8 (3 - Чернавский стр.521, 4- Дунаев стр.457)
- Диаметр вала: 22 мм

```
In [6]: # Диаметр вала электродвигателя
d_ed= 22
# Фактическая частота вращения
```

```
n_fact = 1500 * (100 - 5.8) / 100
print(n_fact)
```

1413.0

```
In [7]: # Передаточное число редуктора
print(f"{n_fact} / {n_out} = {n_fact/n_out:.2f}")
```

1413.0 / 59 = 23.95

```
In [8]: # Принимаю стандартное значение передаточного числа редуктора по таблице 3.2
u_p = 25
```

---

## Определение кинематических и силовых параметров для каждого вала

---

### Быстроходный вал

```
In [9]: # Частота оборотов
n_fact
```

Out[9]: 1413.0

```
In [10]: # Угловая скорость
w1 = pi * n_fact / 30
print(f"{w1:.3f}")
```

147.969

```
In [11]: # Мощность на входном валу после муфты
P1 = 1500 * 0.98
print(P1)
```

1470.0

```
In [12]: # Момент на входном валу, Н*мм
T1 = P1 * 1000 / w1
print(T1)
```

9934.512371341238

---

### Тихоходный вал

```
In [13]: # Частота оборотов
n1 = n_fact
n2 = n1 / u_p
print(n2)
```

56.52

```
In [14]: # Угловая скорость
w2 = pi * n2 / 30
```

```
print(w2)
```

5.91876055936317

```
In [15]: # Мощность на выходном валу
# 0.8 - КПД для червячного редуктора (таблица 1.1)
P2 = P1 * 0.8
print(P2)
```

1176.0

```
In [16]: # Вращающий момент
T2 = P2 * 1000 / w2
print(T2)
```

198690.24742682476

Моменты для дальнейших расчетов

```
In [17]: # Принимаю моменты для расчетов, Н*мм
T1 = 9900
T2 = 198000
```

---

## Выбор материалов и допускаемых напряжений

```
In [18]: # Скорость скольжения
v_s = 4.5 * 10 ** (-4) * n1 * (T2 / 1000) ** (1/3)
print(v_s)
```

3.706036899045438

---

### Выбираю материал для червячного колеса

Согласно таблице 3.1 стр.15

Бр А9ЖЗЛ

$$\sigma_T = 200$$

$$\sigma_B = 400$$

$$\sigma_H = 300 - 25v_s$$

$$\sigma_F = 0,25\sigma_T + 0,08\sigma_B$$

---

$$\sigma_H$$

```
In [19]: po_h = 300 - 25 * v_s
print(f"300 - 25*{v_s:.2f} = {300 - 25*v_s:.1f}")
```

$$300 - 25 \cdot 3.71 = 207.3$$

$$\sigma_F$$

```
In [20]: po_f = 0.25*200 + 0.08*400
print(po_f)
```

82.0

---

Проектный расчет червячной передачи по контактным напряжениям

```
In [21]: # Согласно таблице 3.2 выбираю стандартное передаточное отношение
# Вычисляю число зубьев колеса
z1 = 2
z2 = z1 * u_p
print(z2)
```

50

```
In [22]: # Определяю межосевое расстояние
K_n = 1.1
q = 12.5
T2_m = T2 / 1000
a_w = (z2/q + 1) * ((5400/(z2/q*po_h)) ** 2 * T2_m * K_n) ** (1/3)
print(f"({z2}/{q} + 1) * ((5400/({z2}/{q}*{po_h:.0f})) ** 2 * {T2_m} * {K_n})

# Выбираю стандартное значение a_w = 125 мм
a_w = 125
```

$(50/12.5 + 1) * ((5400/(50/12.5 \cdot 207)) ** 2 * 198.0 * 1.1) ** (1/3) = 104.892$   
331

```
In [23]: # Определяю величину осевого модуля для стандартного a_w
m = 2*a_w/(q + z2)
print(m)
```

4.0

---

Выбираю стандартные значения из таблицы 3.3

$$a_w = 125 \text{ мм}$$

$$m = 4 \text{ мм}$$

$$q = 12,5$$

$$z_2 : z_1 = 50 : 2$$

---

Определяю геометрические параметры червяка

```
In [24]: d1 = d_lb = q * m  
print(d1)
```

50.0

```
In [25]: da1 = d1 + 2 * m  
print(da1)
```

58.0

```
In [26]: df1 = d1 - 2.4*m  
print(df1)
```

40.4

```
In [27]: # Длина нарезанной части червяка  
b1 = (11 + 0.06*z2)*m  
print(b1)
```

56.0

---

## Определяю геометрические параметры колеса

```
In [28]: d2 = d_2t = m * z2  
print(d2)
```

200.0

```
In [29]: da2 = d2 + 2*m  
print(da2)
```

208.0

```
In [30]: # Наибольший диаметр червячного колеса  
d_am2 = da2 + 6*m/(z1 + 2)  
print(f"{da2} + 6*{m}/{(z1} + 2) = {da2 + 6*m/(z1 + 2)}")
```

208.0 + 6\*4.0/(2 + 2) = 214.0

```
In [31]: df2 = d2 - 2.4*m  
print(df2)
```

190.4

```
In [32]: # Ширина зубчатого венца червячного колеса  
b2 = 0.75*da1  
print(b2)  
b2 = 44  
print(b2)
```

43.5

44

Параметры ступицы смотри далее (Конструирование червячной передачи)

```
In [33]: # Проверяю выбранное значение v_s  
tg_y = z1/q
```

```
print(tg_y)
```

0.16

```
In [34]: y = atan(tg_y)
print(y)
```

0.1586552621864014

```
In [35]: # Угол в градусах
deg_y = degrees(y)
print(deg_y)
```

9.090276920822323

```
In [36]: # Нормальный модуль
m_n = m*cos(y)
print(m_n)
```

3.9497625276668216

---

### Проверочный расчет по контактным напряжениям

```
In [37]: # Проверяю условие прочности выбранной стандартной передачи
temp = (5400/(z2/q))*(((z2/q+1)/a_w)**3*T2_m*K_n)**(0.5)
print(f"(5400/({z2}/{q}))*((({z2}/{q}+1)/{a_w}))**3*{T2_m}*{K_n})**(0.5) = {(
```

(5400/(50/12.5))\*(((50/12.5+1)/125)\*\*3\*198.0\*1.1)\*\*(0.5) = 159.38692543618504

Прочность по контрольным напряжениям обеспечена так как 159 МПа < 207 МПа

---

### Проверка прочности зубьев червячного венца на изгиб

```
In [38]: # Окружная сила на колесе
Ft2 = 2 * T2/d2
print(f"2 * {T2}/{d2} = {2 * T2/d2}")
```

2 \* 198000/200.0 = 1980.0

```
In [39]: # Окружная сила на червяке
Ft1 = 2 * T1/d1
print(f"2 * {T1}/{d1} = {2 * T1/d1}")
```

2 \* 9900/50.0 = 396.0

```
In [40]: # Определение коэффициента формы зуба из таблицы 3.7
z_v = z2/cos(y) ** 3
print(f"{z2}/cos({y:.4f}) ** 3 = {z2/cos(y) ** 3:.1f}")
```

50/cos(0.1587) \*\* 3 = 51.9

Принимаю форму зуба из таблицы 3.7

$$Y_F = 1,45$$

Значения коэффициентов расчетной нагрузки  $K_H$  и  $K_F$  принимаются одинаковыми (стр.18) = 1

```
In [41]: Y_F = 1.45
K_F = K_n
po_ff = 0.7*Y_F*Ft2*K_F/(b2*m_n)
print(f"0.7*{Y_F}*{Ft2}*{K_F}/({b2}*{m_n:.2f}) = {0.7*Y_F*Ft2*K_F/(b2*m_n):.1f}")
print(f"{po_ff:.1f} < {po_f}")
print("Прочность по напряжениям изгиба обеспечена")
```

0.7\*1.45\*1980.0\*1.1/(44\*3.95) = 12.72

12.7 < 82.0

Прочность по напряжениям изгиба обеспечена

---

Уточняю КПД червячной передачи

```
In [42]: phi = "1°26'"
f_deg = lambda x: f"{int(x)}°{int((x - int(x))*60)}'"
print(f"tan({f_deg(deg_y)})/tan({f_deg(deg_y)} + {phi})")
kpd_b = tan(y)/tan(y + degrees_to_radians(1, 26))
print(kpd_b)
```

tan(9°5')/tan(9°5' + 1°26')

0.8613018256599354

Общий КПД

КПД = КПДп \* КПДм

```
In [43]: kpd_sum = kpd_b * kpd_m
print(f"{kpd_b:.3f} * {kpd_m} = {kpd_b * kpd_m:.3f}")
# print(kpd_sum)
```

0.861 \* 0.98 = 0.844

Тепловой расчет

```
In [44]: # Тепло выделяемое редуктором
FF = P1*(1 - kpd_b)
print(f"{P1}*(1 - {kpd_b:.2f}) = {P1*(1 - kpd_b):.0f}")
print(FF)
```

1470.0\*(1 - 0.86) = 204

203.88631627989494

```
In [45]: # Площадь редуктора без учета днища
A_plosh = 0.25*0.3 + (0.3*2 + 0.25*2)*0.1
print(f"Площадь редуктора = {0.25*0.3 + (0.3*2 + 0.25*2)*0.1} м2")
t_red = 100 # Температура редуктора допустимая
t_pom = 20 # Температура в помещении
K_tep1 = 17 # Для помещений с интенсивной вентиляцией
# Количество отведенного тепла
```

```
FF_otv = K_templ*(t_red - t_pom)*A_plosh
print(f"Отводимое тепло {FF_otv} Вт")
```

Площадь редуктора = 0.185 м2

Отводимое тепло 251.6 Вт

Параметры ступицы смотри далее (Конструирование червячной передачи)

---

## Расчет валов

Расчет выполняю по таблице 4.1

Быстроходный вал (червяк)

```
In [46]: # 1 ступень под муфту
d_1st = 1.2 * d_ed
print(f"1.2 * {d_ed} = {1.2 * d_ed}")
d_1st = 24
print(d_1st)
```

1.2 \* 22 = 26.4

24

```
In [47]: # 2 ступень под уплотнение крышки с отверстием и подшипник
t = 2 # Таблица 4.2
d_2st = d_1st + 2 * t
print(d_2st)
d_2st = 30
print(d_2st)
```

28

30

```
In [48]: # 3 ступень под шестерню
r = 1 # таблица 4.2
d_3st = d_2st + 3.2 * r
print(d_3st)
d_3st = 34
print(d_3st)
```

33.2

34

```
In [49]: # 4 ступень
d_4st = d_2st
print(d_4st)
```

30

Тихоходной вал (червячное колесо)

```
In [50]: # Выходной вал под муфту: 1-я ступень
d_1st_2 = (T2/(0.2 * 40)) ** (1/3)
print(f"({T2}/(0.2 * 40)) ** (1/3) = {(T2/(0.2 * 40)) ** (1/3):.2f}")
print(d_1st_2)
```



```
d_1st_2 = 30
print(d_1st_2)
```

$(198000 / (0.2 * 40)) ** (1/3) = 29.14$   
29.142383416257275  
30

```
In [51]: # 2-я ступень
t = 2.5
d_2st_2 = d_1st_2 + 2 * t
print(d_2st_2)
```

35.0

```
In [52]: # 3-я ступень
d_3st_2 = d_2st_2 + 3.2 * r
print(f"{d_2st_2} + 3.2 * {r} = {d_2st_2 + 3.2 * r}")
print(d_3st_2)
d_3st_2 = 40
print(d_3st_2)
```

$35.0 + 3.2 * 1 = 38.2$   
38.2  
40

```
In [53]: # 4-я ступень
d_4st_2 = d_2st_2
print(d_4st_2)
```

35.0

---

## Конструирование червячной передачи стр.20 (Ступица)

```
In [54]: # Наибольший диаметр червячного колеса
d_am2 = da2 + 6*m/(z1 + 2)
print(f"{da2} + 6*{m}/({z1} + 2) = {da2 + 6*m/(z1 + 2)}")
```

$208.0 + 6*4.0/(2 + 2) = 214.0$

```
In [55]: # Наружный диаметр ступицы
d_st = 1.8 * d_3st_2
print(f"1.8 * {d_3st_2} = {1.8 * d_3st_2}")
```

$1.8 * 40 = 72.0$

```
In [56]: # Ширина ступицы
l_st = 1.7 * d_3st_2
print(l_st)
```

68.0

```
In [57]: po_1 = po_2 = 2 * m
print(po_1)
```

8.0

```
In [58]: C = 0.25 * b2
print(C)
C = 12
print(C)
```

```
11.0
12
```

```
In [59]: # d винта
d_vint = 1.4*m
print(d_vint)
```

```
5.6
```

```
In [60]: # l винта
l_vint = 0.4*b2
print(l_vint)
```

```
17.6
```

```
In [61]: f = 0.2*d_vint
print(f)
```

```
1.1199999999999999
```

## Параметры ступеней валов и подшипников (табл. 4.6)

```
In [62]: from IPython.display import Markdown
```

```
l_1st = 50
l_2st = 55
l_3st = 170
l_4st = 30
```

```
l_1st_2 = 60
l_2st_2 = 45
l_3st_2 = 80
l_4st_2 = 20
```

```
Markdown(f"""
```

```
##### Быстроходный вал
```

```
| d1 / l1 | d2 / l2 | d3 / l3 | d4 / l4 |
|-----|-----|-----|-----|
| {d_1st} / {l_1st} | {d_2st} / {l_2st} | {d_3st} / {l_3st} | {d_4st} / {l_4st} |
```

```
##### Подшипники для быстроходного вала
```

```
| Обозначение | $d \times B(T)$ | $C_r$, кН$ | $C_0$, кН$ | $\alpha$ $ |
|-----|-----|-----|-----|-----|
| 7606 | 30x72x29x29 | 63 | 51 | 12 |
***
```

```
##### Тихоходный вал
```

```
| d1 / l1 | d2 / l2 | d3 / l3 | d4 / l4 |
|-----|-----|-----|-----|
| {d_1st_2} / {l_1st_2} | {d_2st_2} / {l_2st_2} | {d_3st_2} / {l_3st_2} | {d_4st_2} / {l_4st_2} |
```

```
##### Подшипники для тихоходного вала
```

```
| Обозначение | $dxDxB(T)$ | $Cr, кН$ | $C_0, кН$ | $α$ |
|-----|-----|-----|-----|-----|
| 36207 | 35x72x17 | 24 | 18 | 12 |
***
""")
```

Out[62]: Быстроходный вал

| d1 / l1 | d2 / l2 | d3 / l3  | d4 / l4 |
|---------|---------|----------|---------|
| 24 / 50 | 30 / 55 | 34 / 170 | 30 / 30 |

Подшипники для быстроходного вала

| Обозначение | $dxDxB(T)$  | $Cr,$<br>кН | $C_0,$<br>кН | $\alpha$<br>° |
|-------------|-------------|-------------|--------------|---------------|
| 7606        | 30x72x29x29 | 63          | 51           | 12            |

Тихоходный вал

| d1 / l1 | d2 / l2   | d3 / l3 | d4 / l4   |
|---------|-----------|---------|-----------|
| 30 / 60 | 35.0 / 45 | 40 / 80 | 35.0 / 20 |

Подшипники для тихоходного вала

| Обозначение | $dxDxB(T)$ | $Cr,$<br>кН | $C_0,$<br>кН | $\alpha$<br>° |
|-------------|------------|-------------|--------------|---------------|
| 36207       | 35x72x17   | 24          | 18           | 12            |

## Выбор шпонок

Шпонки выбираю по таб. 4.8 стр 23

Быстроходный вал: шпонка

```
In [63]: # Быстроходный вал: хвостовик
# Проверяю шпонку на смятие
b_sh_b = 8
h_sh_b = 7
l_obsh_b = 32
l_rb = l_obsh_b - b_sh_b
print(l_rb)
t1_b = 4
p_sm = 2 * T1 / (d_1st * l_rb * (h_sh_b - t1_b))
print(f"смятие = 2 * {T1} / ({d_1st} * {l_rb} * ({h_sh_b} - {t1_b})) = {2 * T1 / (d_1st * l_rb * (h_sh_b - t1_b))}")
```

24

$\text{смятие} = 2 * 9900 / (24 * 24 * (7 - 4)) = 11.458333333333334$

```
In [64]: # Тихоходный вал: под колесом
# Проверяю шпонку на смятие
b_sh_t = 12
h_sh_t = 8
l_obsh_t = 50
l_rt = l_obsh_t - b_sh_t
t1_t = 5
p_sm = 2 * T2 / (d_3st_2 * l_rt * (h_sh_t - t1_t))
print(f"2 * {T2} / ({d_3st_2} * {l_rt} * ({h_sh_t} - {t1_t})) = {2 * T2 / (d_3st_2 * l}

2 * 198000 / (40 * 38 * (8 - 5)) = 86.84210526315789
```

```
In [65]: # Тихоходный вал: хвостовик
# Проверяю шпонку на смятие
b_sh_out = 10
h_sh_out = 8
l_obsh_out = 56
l_r_out = l_obsh_out - b_sh_out
t1_t_out = 5
p_sm = 2 * T2 / (d_1st_2 * l_r_out * (h_sh_out - t1_t_out))
print(f"2 * {T2} / ({d_1st_2} * {l_r_out} * ({h_sh_out} - {t1_t_out})) = {2 * T2 / (}

2 * 198000 / (30 * 46 * (8 - 5)) = 95.7
```

---

## Определение сил в зацеплении и консольных сил от муфт

табл.4.10

---

```
In [66]: # Окружная сила на червяке
print(f"2 * {T1} / {d1} = {2 * T1 / d1}")
print("Ft1")
```

$2 * 9900 / 50.0 = 396.0$

Ft1

```
In [67]: # Окружная сила на колесе
print(f"2 * {T2} / {d2} = {2 * T2 / d2}")
print("Ft2")
```

$2 * 198000 / 200.0 = 1980.0$

Ft2

```
In [68]: # Радиальная сила на колесе и на червяке
Fr2 = Ft2 * tan(radians(20))
print(Fr2)
Fr1 = Fr2 = 720
print(Fr2)
```

720.6610638470806

720

```
In [69]: # Осевая сила на червяке
Fa1 = Ft2
print(Fa1)
```

1980.0

```
In [70]: # Осевая сила на колесе
Fa2 = Ft1
print(Fa2)
```

396.0

```
In [71]: # Сила на муфте (быстроходный вал)
Fm1 = 4*T1 ** 0.5
print(Fm1)
Fm1 = 398
print(Fm1)
```

397.994974842648

398

```
In [72]: # Сила на муфте (тихоходный вал)
Fm2 = 4*T2 ** 0.5
print(Fm2)
Fm2 = 1780
print(Fm2)
```

1779.887636902959

1780

---

## Реакции и эпюры

---

### Расчет быстроходного вала

```
In [73]: # Расчет места приложения сил в радиально упорном подшипнике
print(0.5*(29+0.5*(30+72)*tan(radians(12))))
```

19.920192322585564

```
In [74]: # Расчет быстроходного вала
def calc_b():
    # Подшипник: 7606
    d_podsh = 30
    D_podsh = 72
    T_podsh = 29
    alpha_podsh = 12

    # Точка приложения опорной силы в коническом подшипнике
    a_podsh = 0.5*(T_podsh + 0.5*(d_podsh + D_podsh)*tan(radians(alpha_podsh)))
    print(f"a_podsh = {a_podsh}")
    a_podsh = 20
```

```

print(f"a_podsh = {a_podsh}")

# Списки для построения эпюр
lst_x1 = [0]
lst_M1 = [0]
lst_x2 = [0]
lst_M2 = [0]

# Червяк: длины, мм
l1 = l_1st
l2 = l_2st
l3 = l_3st
l4 = l_4st
d1 = d_1b

# Участки, мм
a = T_podsh - a_podsh + l3/2
b = a
c = l2 - (T_podsh - a_podsh) + l1/2
d = d1/2
print(f"a = {a}, b = {b}, c = {c}, d = {d}")

# Силы, Н
Ft = Ft1
Fr = Fr1
Fa = Fa1
Fm = Fm1

#-----
# Вертикальная плоскость: XY

#-----1. Реакции-----
# Моменты: положительно = против часовой стрелки

#  $\sum M(A) = 0$ 
#  $-Fr*a - Fa*d + Rb*(a + b) = 0$ 
Rb = Rb1 = (Fr*a + Fa*d)/(a + b)

#  $\sum M(B) = 0$ 
#  $-Ra*(a + b) + Fr*b - Fa*d = 0$ 
Ra = Ra1 = (Fr*b - Fa*d)/(a + b)

# Проверка ось Y
#  $Ra + Rb - Fr = 0$ 
assert Ra + Rb - Fr == 0, 'Сумма сил должна быть равна нулю'

# Вывод результата
print("\n1.Расчет реакции опор")
print("Вертикальная плоскость: XY")
print(f"Момент Fa*d = {Fa*d} Н·мм")
print("Ось Y")
print(f"Ra = {Ra:.3f}")
print(f"Rb = {Rb:.3f}\n")

#-----2. Эпюры-----

```

```

# Силы: положительно = балка крутится
# по часовой стрелке
#
# Моменты: положительно = балка хочет изогнуться
# вверх

# 1.Участок  $0 \leq x \leq a$ 
#  $x = a$ 
Ma = Ra * a

# Записываю результат
lst_x1 += [a]
lst_M1 += [Ma]

# Вывод результата
print("\n2.Построение эпюр")
print("Вертикальная плоскость: XY")
print("1.Участок  $0 \leq x \leq a$ ")
print("x = 0: Mxy = 0")
print(f"x = a: Mxy = {Ma:.1f}\n")

# 2.Участок  $a \leq x \leq (a + b)$ 
#  $x = a$ 
Ma = Ra * a + Fa * d
#  $x = a + b$ 
Mb = Ra * (a + b) + Fa * d - Fr * b

# Записываю результат
lst_x1 += [a, a + b, a + b + c]
lst_M1 += [Ma, Mb, 0]

# Вывод результата
print("2.Участок  $a \leq x \leq (a + b)$ ")
print(f"x = a: Mxy = {Ma:.1f}")
print(f"x = a+b: Mxy = {Mb:.1f}\n")
print("-----\n")

#-----
# Горизонтальная плоскость: XZ

#-----1. Реакции-----
# Моменты: положительно = против часовой стрелки

#  $\sum M(A) = 0$ 
#  $Ft*a - Rb*(a + b) - Fm*(a + b + c) = 0$ 
Rb = Rb2 = (Ft*a - Fm*(a + b + c))/(a + b)

#  $\sum M(B) = 0$ 
#  $Ra*(a + b) - Ft*b - Fm*c = 0$ 
Ra = Ra2 = (Ft*b + Fm*c)/(a + b)

# Проверка ось Z
#  $Ra - Ft + Rb + Fm = 0$ 

```

```
assert Ra - Ft + Rb + Fm == 0 , 'Сумма сил должна быть равна нулю'
```

```
# Вывод результата
```

```
print("1.Расчет реакции опор")  
print("Горизонтальная плоскость: XZ")  
print("Ось Z")  
print(f"Ra = {Ra:.3f}")  
print(f"Rb = {Rb:.3f}\n")
```

```
#-----2. Эпюры-----
```

```
# Силы: положительно = балка крутится  
# по часовой стрелке  
#  
# Моменты: положительно = балка хочет изогнуться  
# вверх
```

```
# 1.Участок  $0 \leq x \leq a$   
#  $x = a$   
Ma = -Ra * a
```

```
# Записываю результат  
lst_x2 += [a]  
lst_M2 += [Ma]
```

```
# Вывод результата  
print("\n2.Построение эпюр")  
print("Горизонтальная плоскость: XZ\n")  
print("1.Участок  $0 \leq x \leq a$ ")  
print("x = 0: Миу = 0")  
print(f"x = a: Миу = {Ma:.1f}\n")
```

```
# 2.Участок  $a \leq x \leq (a + b)$   
#  $x = a$   
Ma = -Ra * a  
#  $x = a + b$   
Mb = -Ra * (a + b) + Ft * b
```

```
# Записываю результат  
lst_x2 += [a, a + b]  
lst_M2 += [Ma, Mb]
```

```
# Вывод результата  
print("2.Участок  $a \leq x \leq (a + b)$ ")  
print(f"x = a: Миу = {Ma:.1f}")  
print(f"x = a+b: Миу = {Mb:.1f}\n")
```

```
# 3.Участок  $(a + b) \leq x \leq (a + b + c)$   
#  $x = a + b$   
Mb = -Ra * (a + b) + Ft * b  
#  $x = a + b + c$   
Mc = -Ra * (a + b + c) + Ft * (b + c) - Rb * c
```

```
# Записываю результат  
lst_x2 += [a + b, a + b + c]  
lst_M2 += [Mb, Mc]
```



```

# Вывод результата
print("3.Участок  $(a + b) \leq x \leq (a + b + c)$ ")
print(f"x = a+b: Миу = {Mb:.1f}")
print(f"x = a+b+c: Миу = {Mc:.1f}\n")

return (lst_x1, lst_M1), (lst_x2, lst_M2), (Ra1 ** 2 + Ra2 ** 2) ** 0.5,
resultY, resultX, Ra_sum_1, Rb_sum_1 = calc_b()

```

a\_podsh = 19.920192322585564  
a\_podsh = 20  
a = 94.0, b = 94.0, c = 71.0, d = 25.0

1.Расчет реакции опор  
Вертикальная плоскость: XY  
Момент  $Fa \cdot d = 49500.0$  Н·мм  
Ось Y  
Ra = 96.702  
Rb = 623.298

2.Построение эпюр  
Вертикальная плоскость: XY  
1.Участок  $0 \leq x \leq a$   
x = 0: Миу = 0  
x = a: Миу = 9090.0

2.Участок  $a \leq x \leq (a + b)$   
x = a: Миу = 58590.0  
x = a+b: Миу = 0.0

-----

1.Расчет реакции опор  
Горизонтальная плоскость: XZ  
Ось Z  
Ra = 348.309  
Rb = -350.309

2.Построение эпюр  
Горизонтальная плоскость: XZ

1.Участок  $0 \leq x \leq a$   
x = 0: Миу = 0  
x = a: Миу = -32741.0

2.Участок  $a \leq x \leq (a + b)$   
x = a: Миу = -32741.0  
x = a+b: Миу = -28258.0

3.Участок  $(a + b) \leq x \leq (a + b + c)$   
x = a+b: Миу = -28258.0  
x = a+b+c: Миу = -0.0

---

## Суммарная реакция опор на подшипниках быстроходного вала

```
In [75]: # Подшипник A
print(f"Ra_сум = {Ra_sum_1:.1f} Н\n")
# Подшипник B
print(f"Rb_сум = {Rb_sum_1:.1f} Н\n")
```

Ra\_сум = 361.5 Н

Rb\_сум = 715.0 Н

## Максимальный изгибающий момент для быстроходного вала

```
In [76]: M_k1 = (58590 ** 2 + 32741 ** 2) ** 0.5
print(int(M_k1))
M_k1 = 67100
print(M_k1)
```

67117

67100

---

## Графики изгибающих моментов для быстроходного вала

```
In [77]: import matplotlib.pyplot as plt

x, M = resultY

plt.figure(figsize=(8, 4))
plt.plot(x, M, marker='o', linestyle='-', color='blue', label='Эпюра Миу(x)')
plt.axhline(0, color='black', linewidth=0.5)
plt.xlabel('Длина вала (мм)')
plt.ylabel('Момент (Н·мм)')
plt.title('Вертикальная плоскость')
plt.grid(True)
plt.legend()

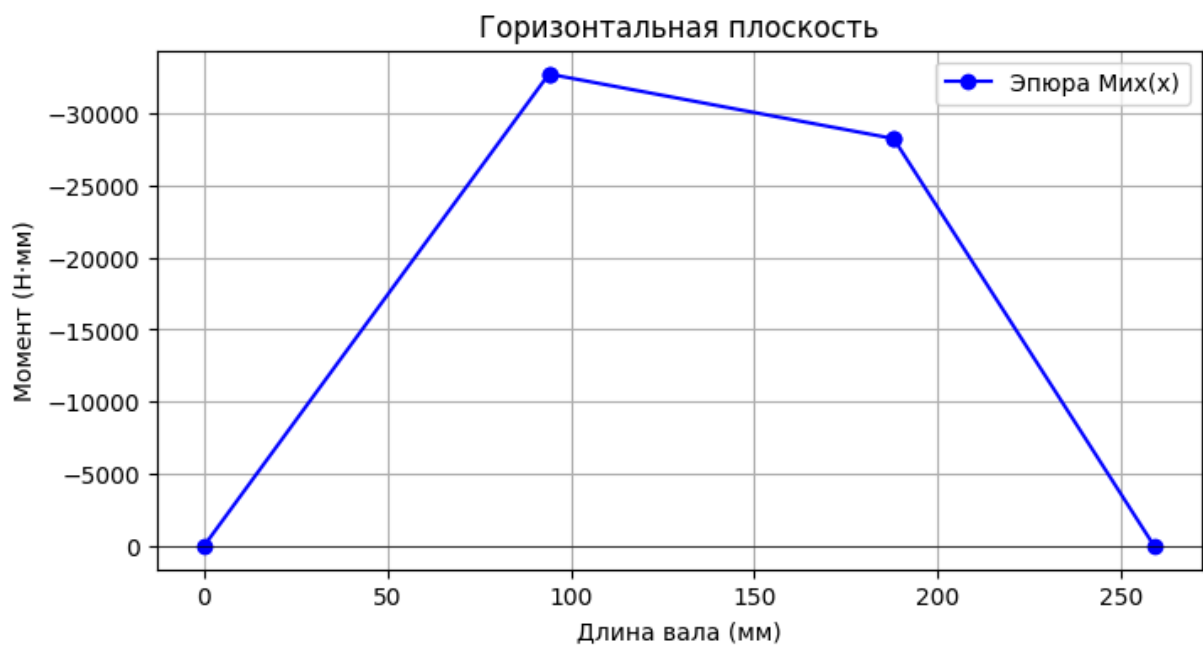
# Инвертируем ось Y
plt.gca().invert_yaxis()
plt.show()
```



In [78]: `x, M = resultX`

```
plt.figure(figsize=(8, 4))
plt.plot(x, M, marker='o', linestyle='-', color='blue', label='Эпюра Мих(x)')
plt.axhline(0, color='black', linewidth=0.5)
plt.xlabel('Длина вала (мм)')
plt.ylabel('Момент (Н·мм)')
plt.title('Горизонтальная плоскость')
plt.grid(True)
plt.legend()

# Инвертируем ось Y
plt.gca().invert_yaxis()
plt.show()
```



## Расчет тихоходного вала

In [79]: **def** calc\_t():

```
# Подшипник: 36207
d_podsh = 35
D_podsh = 72
B_podsh = 17
alpha_podsh = 12

# Точка приложения опорной силы в радиально-упорном подшипнике
a_podsh = 0.5*(B_podsh + 0.5*(d_podsh + D_podsh)*tan(radians(alpha_podsh)))
print(f"a_podsh = {a_podsh}")
a_podsh = 14
print(f"a_podsh = {a_podsh}")

# Списки для построения эпюр
lst_x1 = [0]
lst_M1 = [0]
lst_x2 = [0]
lst_M2 = [0]

# Тихоходный вал: длины, мм
l1 = l_1st_2
l2 = l_2st_2
l3 = l_3st_2
l4 = l_4st_2
d1 = d_2t

# Участки, мм
a = B_podsh - a_podsh + l3/2
b = a
c = l2 - (B_podsh - a_podsh) + l1/2
d = d1/2
print(f"a = {a}, b = {b}, c = {c}, d = {d}")

# Силы, Н
Ft = Ft2
Fr = Fr2
Fa = Fa2
Fm = Fm2

#-----
# Вертикальная плоскость: XY

#-----1. Реакции-----
# Моменты: положительно = против часовой стрелки

# ΣM(A) = 0
# -Fr*a + Fa*d + Rb*(a + b) = 0
Rb = Rb1 = (Fr*a - Fa*d)/(a + b)

# ΣM(B) = 0
```

```

#  $-Ra*(a + b) + Fr*b + Fa*d = 0$ 
Ra = Ra1 = (Fr*b + Fa*d)/(a + b)

# Проверка ось Y
#  $Ra + Rb - Fr = 0$ 
assert Ra + Rb - Fr == 0, 'Сумма сил должна быть равна нулю'

# Вывод результата
print("\n1.Расчет реакции опор")
print("Вертикальная плоскость: XY")
print(f"Момент Fa*d = {Fa*d} Н·мм")
print("Ось Y")
print(f"Ra = {Ra:.3f}")
print(f"Rb = {Rb:.3f}\n")

#-----2. Эпюры-----
# Силы: положительно = балка крутится
# по часовой стрелке
#
# Моменты: положительно = балка хочет изогнуться
# вверх

# 1.Участок  $0 \leq x \leq a$ 
#  $x = a$ 
Ma = Ra * a

# Записываю результат
lst_x1 += [a]
lst_M1 += [Ma]

# Вывод результата
print("\n2.Построение эпюр")
print("Вертикальная плоскость: XY")
print("1.Участок  $0 \leq x \leq a$ ")
print("x = 0: Миу = 0")
print(f"x = a: Миу = {Ma:.1f}\n")

# 2.Участок  $a \leq x \leq (a + b)$ 
#  $x = a$ 
Ma = Ra * a - Fa * d
#  $x = a + b$ 
Mb = Ra*(a + b) - Fa * d - Fr * b

# Записываю результат
lst_x1 += [a, a + b, a + b + c]
lst_M1 += [Ma, Mb, 0]

# Вывод результата
print("2.Участок  $a \leq x \leq (a + b)$ ")
print(f"x = a: Миу = {Ma:.1f}")
print(f"x = a+b: Миу = {Mb:.1f}\n")
print("-----\n")

```

```

#-----
# Горизонтальная плоскость: XZ

#-----1. Реакции-----
# Моменты: положительно = против часовой стрелки

#  $\sum M(A) = 0$ 
#  $-F_t \cdot a + R_b \cdot (a + b) + F_m \cdot (a+b+c) = 0$ 
Rb = Rb2 = (Ft*a - Fm*(a+b+c))/(a + b)

#  $\sum M(B) = 0$ 
#  $-R_a \cdot (a + b) + F_t \cdot b + F_m \cdot c = 0$ 
Ra = Ra2 = (Ft*b + Fm*c)/(a + b)

# Проверка
assert Ra - Ft + Rb + Fm == 0 , 'Сумма сил должна быть равна нулю'

# Вывод результата
print("1.Расчет реакции опор")
print("Горизонтальная плоскость: XZ")
print(f"Ra = {Ra:.3f}")
print(f"Rb = {Rb:.3f}\n")

#-----2. Эпюры-----
# Силы: положительно = балка крутится
# по часовой стрелке
#
# Моменты: положительно = балка хочет изогнуться
# вверх

# 1.Участок  $0 \leq x \leq a$ 
#  $x = a$ 
Ma = Ra * a

# Записываю результат
lst_x2 += [a]
lst_M2 += [Ma]

# Вывод результата
print("\n2.Построение эпюр")
print("Горизонтальная плоскость: XZ\n")
print("1.Участок  $0 \leq x \leq a$ ")
print("x = 0: Mиз = 0")
print(f"x = a: Mиз = {Ma:.1f}\n")

# 2.Участок  $a \leq x \leq (a + b)$ 
#  $x = a$ 
Ma = Ra * a
#  $x = a + b$ 
Mb = Ra * (a + b) - Ft * b

# Записываю результат
lst_x2 += [a, a + b, a + b + c]
lst_M2 += [Ma, Mb, Ra*(a+b+c) - Ft*(b+c) + Rb*c]

```

```

# Вывод результата
print("2.Участок  $a \leq x \leq (a + b)$ ")
print(f"x = a: Миу = {Ma:.1f}")
print(f"x = a+b: Миу = {Mb:.1f}\n")
return (lst_x1, lst_M1), (lst_x2, lst_M2), (Ra1 ** 2 + Ra2 ** 2) ** 0.5,
resultY_2, resultX_2, Ra_sum_2, Rb_sum_2 = calc_t()

```

```

a_podsh = 14.185888024673092
a_podsh = 14
a = 43.0, b = 43.0, c = 72.0, d = 100.0

```

```

1.Расчет реакции опор
Вертикальная плоскость: XY
Момент Fa*d = 39600.0 Н·мм
Ось Y
Ra = 820.465
Rb = -100.465

```

```

2.Построение эпюр
Вертикальная плоскость: XY
1.Участок  $0 \leq x \leq a$ 
x = 0: Миу = 0
x = a: Миу = 35280.0

```

```

2.Участок  $a \leq x \leq (a + b)$ 
x = a: Миу = -4320.0
x = a+b: Миу = 0.0

```

-----

```

1.Расчет реакции опор
Горизонтальная плоскость: XZ
Ra = 2480.233
Rb = -2280.233

```

```

2.Построение эпюр
Горизонтальная плоскость: XZ

```

```

1.Участок  $0 \leq x \leq a$ 
x = 0: Миу = 0
x = a: Миу = 106650.0

```

```

2.Участок  $a \leq x \leq (a + b)$ 
x = a: Миу = 106650.0
x = a+b: Миу = 128160.0

```

---

Суммарная реакция опор на подшипниках тихоходного вала

```

In [80]: # Подшипник A
print(f"Ra_сум = {Ra_sum_2:.1f} Н\n")

```

```
# Подшипник B
print(f"Rb_сум = {Rb_sum_2:.1f} Н\n")
```

Ra\_сум = 2612.4 Н

Rb\_сум = 2282.4 Н

## Максимальный изгибающий момент для тихоходного вала

```
In [81]: M_k12 = (35280 ** 2 + 106640 ** 2) ** 0.5
print(M_k12)
M_k22 = 128160 # Самый большой момент!!!
M_k2 = 128160
print(M_k2)
```

112324.38737869884

128160

---

## Графики изгибающих моментов для тихоходного вала

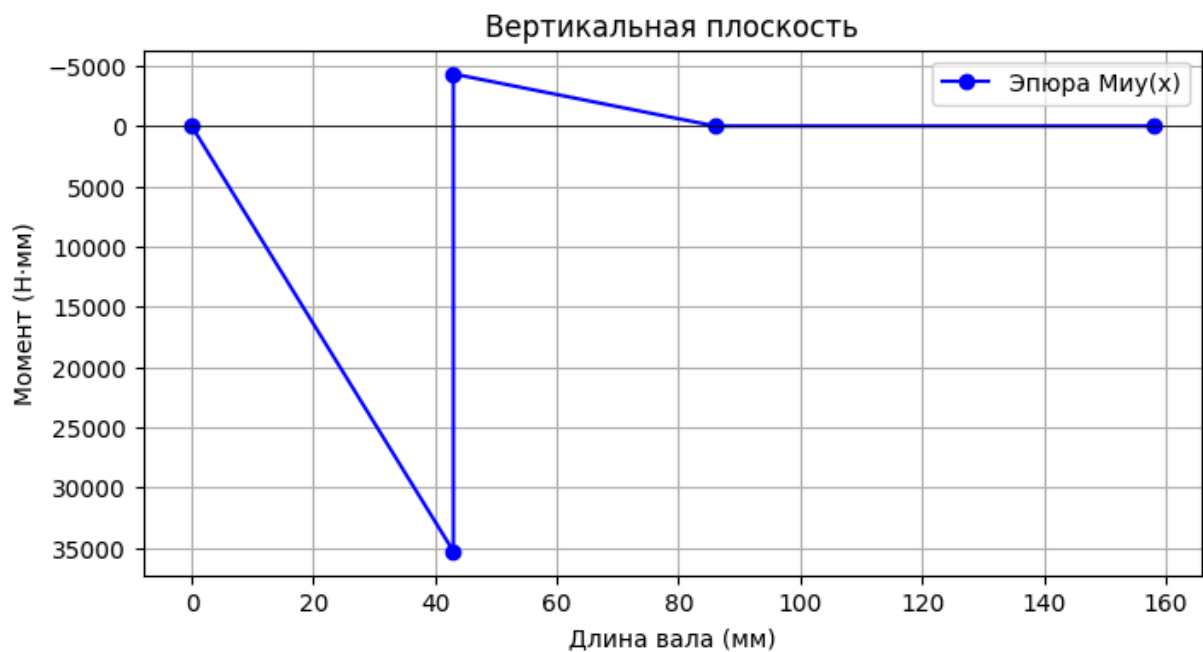
```
In [82]: import matplotlib.pyplot as plt

x, M = resultY_2

plt.figure(figsize=(8, 4))
plt.plot(x, M, marker='o', linestyle='-', color='blue', label='Эпюра Миу(x)')
plt.axhline(0, color='black', linewidth=0.5)
plt.xlabel('Длина вала (мм)')
plt.ylabel('Момент (Н·мм)')
plt.title('Вертикальная плоскость')
plt.grid(True)
plt.legend()

# Инвертируем ось Y
plt.gca().invert_yaxis()
plt.show()
```

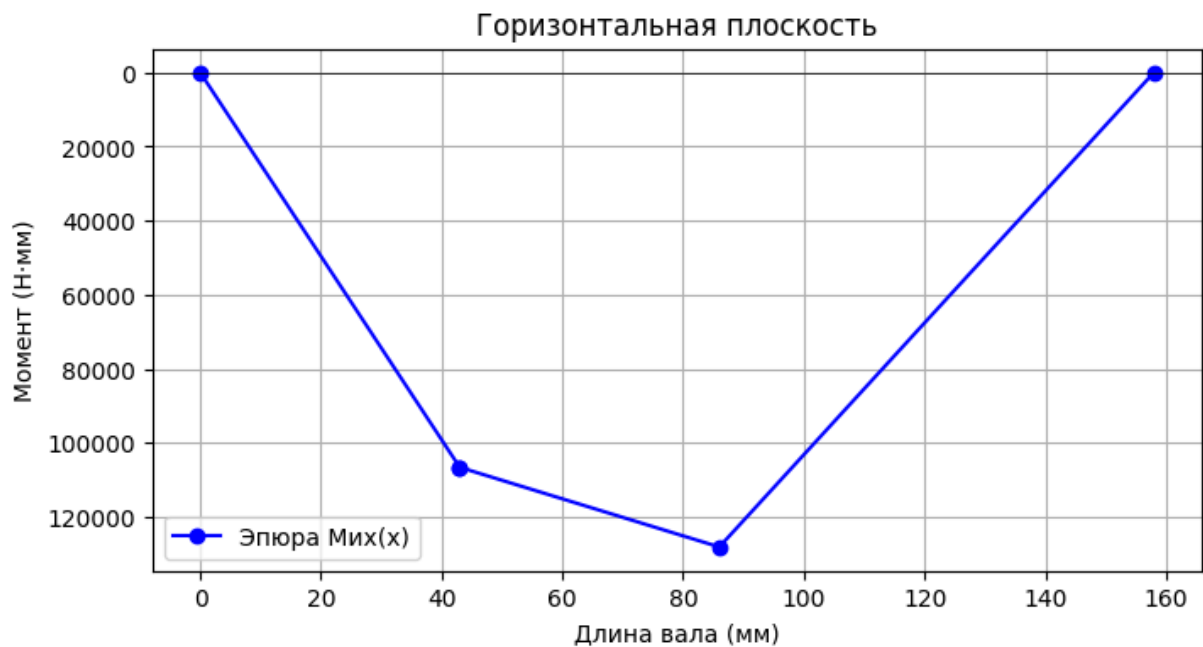




```
In [83]: x, M = resultX_2

plt.figure(figsize=(8, 4))
plt.plot(x, M, marker='o', linestyle='-', color='blue', label='Эпюра Мих(x)')
plt.axhline(0, color='black', linewidth=0.5)
plt.xlabel('Длина вала (мм)')
plt.ylabel('Момент (Н·мм)')
plt.title('Горизонтальная плоскость')
plt.grid(True)
plt.legend()

# Инвертируем ось Y
plt.gca().invert_yaxis()
plt.show()
```



---

## Проверка валов на усталостную прочность

```
In [84]: # Считаю коэффициент для углеродистой стали
# po_b = 900 (3 - Чернавский стр 88)
po_b = 900
po__1 = 0.43*po_b
print(po__1)
```

387.0

```
In [85]: t__1 = 0.58*po__1
print(int(t__1))
```

224

```
In [86]: W_netto = 0.1 * df1 ** 3
print(W_netto)
W_knetto = 0.2*df1 ** 3
print(W_knetto)
po_a = M_k1/W_netto
print(po_a)
```

6593.9264

13187.8528

10.176031082178897

```
In [87]: k_po = 2.45
k_f = 0.9
k_t = 2
k_d = 0.85
t_a = 0.5*T1/W_knetto
ksi = 0.1
ksi_t = 0.05
po_m = 20
t_m = 40
print(t_a)
```

0.3753454087688937

```
In [88]: s_po = po__1/(k_po*po_a/(k_d*k_f) + ksi*po_m)
print(f"{po__1}/({k_po}*{po_a}/({k_d}*{k_f}) + {ksi}*{po_m}) = {po__1/(k_po*")
print(s_po)
```

387.0/(2.45\*10.176031082178897/(0.85\*0.9) + 0.1\*20) = 11.188235907701179

11.188235907701179

```
In [89]: s_t = t__1/(k_t*t_a/(k_d*k_f) + ksi_t*t_m)
print(f"{t__1}/({k_t}*{t_a}/({k_d}*{k_f}) + {ksi_t}*{t_m}) = {t__1/(k_t*t_a/")
print(s_t)
```

224.45999999999998/(2\*0.3753454087688937/(0.85\*0.9) + 0.05\*40) = 75.28942488

810411

75.28942488810411

```
In [90]: s_1 = s_po * s_t/(s_po ** 2 + s_t ** 2) ** 0.5  
print(s_1)
```

11.066710849766016

Для быстроходного вала

Вывод: 10,58 > [s = 2,5..3]

```
In [91]: # Тихоходный вал  
# W_netto2 = 0.1 * d_3st_2 ** 3  
W_netto2 = 0.1 * d_2st_2 ** 3          # Изгибающий момент на подшипнике B  
print(W_netto2)  
W_knetto2 = 0.2*d_2st_2 ** 3  
print(W_knetto2)  
po_a2 = M_k2/W_netto2  
print(po_a2)
```

4287.5

8575.0

29.891545189504374

```
In [92]: t_a2 = 0.5*T2/W_knetto2  
print(t_a2)
```

11.545189504373178

```
In [93]: s_po2 = po_1/(k_po*po_a2/(k_d*k_f) + ksi*po_m)  
print(f"{po_1}/({k_po}*{po_a2}/({k_d}*{k_f}) + {ksi}*{po_m}) = {po_1/(k_po  
print(s_po2)
```

387.0/(2.45\*29.891545189504374/(0.85\*0.9) + 0.1\*20) = 3.9598452278589855

3.9598452278589855

```
In [94]: s_t2 = t_1/(k_t*t_a2/(k_d*k_f) + ksi_t*t_m)  
print(f"{t_1}/({k_t}*{t_a2}/({k_d}*{k_f}) + {ksi_t}*{t_m}) = {t_1/(k_t*t_a  
print(s_t2)
```

224.45999999999998/(2\*11.545189504373178/(0.85\*0.9) + 0.05\*40) = 6.974380854

94133

6.97438085494133

```
In [95]: s_2 = s_po2 * s_t2/(s_po2 ** 2 + s_t2 ** 2) ** 0.5  
print(s_2)
```

3.443522122751513

Для тихоходного вала

Вывод: 4,83 > [s = 2,5..3]

---

## Проверка подшипников по динамической грузоподъемности

### Расчет подшипников на быстроходном валу

```
In [96]: print(f"Fa1/VFr_A = {Fa1/Ra_sum_1:.1f}")
print(f"Fa1/VFr_B = {Fa1/Rb_sum_1:.1f}")
```

Fa1/VFr\_A = 5.5  
Fa1/VFr\_B = 2.8

```
In [97]: def resurs1():

    # Частота оборотов
    n = n1

    # Подшипник: данные
    # 7606
    Cr = 63000
    e = 0.319
    Y = 1.882
    X = 0.4

    V = 1
    Kb = 1.3
    Kt = 1
    p = 10/3

    # -----Силы-----
    #
    #          (Fa)--->
    #   (A)--->      <--- (B)
    #
    # Силы от подшипников и осевая сила
    # направлены как на рисунке: Vab == True
    #
    # Если сила Fa направлена в другую сторону
    # Установить флаг Vab == false или указать
    # силу со знаком минус

    Vab = True

    # Силы: указывать в Ньютонах
    Fa = Fa1
    Ra = Ra_sum_1
    Rb = Rb_sum_1

    print("Входные данные")
    print("\nСилы:")
    print(f"Fa = {Fa}")
    print(f"Ra = {Ra}")
    print(f"Rb = {Rb}")

    print("\nПодшипник:")
    print(f"Cr = {Cr}")
    print(f"e = {e}")
    print(f"X = {X}")
    print(f"Y = {Y}")

    print("\nПостоянные:")
```

```

print(f"p = {p:.3f}")
print(f"V = {V}")
print(f"Kb = {Kb}")
print(f"Kt = {Kt}")

if Fa < 0:
    Vab = False
    Fa = -Fa

S1 = 0.83 * e * Ra
S2 = 0.83 * e * Rb

if Vab:
    F1 = S1 if (-S1 - Fa + S2) < 0 else S2 - Fa
    F2 = S2 if (S1 + Fa - S2) < 0 else S1 + Fa
else:
    F1 = S1 if (-S1 + Fa + S2) < 0 else S2 + Fa
    F2 = S2 if (S1 - Fa - S2) < 0 else S1 - Fa

f_ek = lambda r, a: (X*V*r + Y*a)*Kb*Kt
F1_ek = f_ek(Ra, F1)
F2_ek = f_ek(Rb, F2)

print("\nВыходные данные")
print(f"Fa = {Fa}")
print(f"Ra = {Ra:.1f}")
print(f"Rb = {Rb:.1f}")
print(f"S1 = {S1:.1f}")
print(f"S2 = {S2:.1f}")
print(f"F1 = {F1:.1f}")
print(f"F2 = {F2:.1f}")
print(f"Fa1/VFr1 = {F1/Ra:.1f}")
print(f"Fa2/VFr2 = {F2/Rb:.1f}")
print(f"F1_ek = {F1_ek:.1f}")
print(f"F2_ek = {F2_ek:.1f}")

f_l = lambda x: (Cr/x) ** p
L1 = f_l(F1_ek)
L2 = f_l(F2_ek)

print(f"Срок службы 1-го подшипника = {L1:.0f} млн.об")
print(f"Срок службы 2-го подшипника = {L2:.0f} млн.об")

f_lh = lambda x: 1e6 * x / (60 * n)
lh1 = f_lh(L1)
lh2 = f_lh(L2)

print(f"Срок службы 1-го подшипника = {lh1:.0f} часов")
print(f"Срок службы 2-го подшипника = {lh2:.0f} часов")

```

resurs1()

## Входные данные

Силы:

$$F_a = 1980.0$$

$$R_a = 361.4832224003735$$

$$R_b = 714.9939092675712$$

Подшипник:

$$C_r = 63000$$

$$e = 0.319$$

$$X = 0.4$$

$$Y = 1.882$$

Постоянные:

$$p = 3.333$$

$$V = 1$$

$$K_b = 1.3$$

$$K_t = 1$$

## Выходные данные

$$F_a = 1980.0$$

$$R_a = 361.5$$

$$R_b = 715.0$$

$$S_1 = 95.7$$

$$S_2 = 189.3$$

$$F_1 = 95.7$$

$$F_2 = 2075.7$$

$$F_{a1}/V_{Fr1} = 0.3$$

$$F_{a2}/V_{Fr2} = 2.9$$

$$F_{1\_ek} = 422.1$$

$$F_{2\_ek} = 5450.2$$

$$\text{Срок службы 1-го подшипника} = 17631805 \text{ млн.об}$$

$$\text{Срок службы 2-го подшипника} = 3492 \text{ млн.об}$$

$$\text{Срок службы 1-го подшипника} = 207971283 \text{ часов}$$

$$\text{Срок службы 2-го подшипника} = 41190 \text{ часов}$$

Подшипники быстроходного вала проработают дольше указанного в задании срока  $44092 > 16000$  часов

---

## Расчет подшипников на тихоходном валу

Устанавливаю радиально-упорные подшипники для схемы установки вала "враспор". Для этого выбираю минимально-возможный подшипник "36207"

Подшипник нахожу в 3-Чернавский стр.435 (или 219 дежавю): **36207** ( $\alpha = 12^\circ$ )

Определяю  $e$

```
In [98]: C_0_podsh_t = 18100
print(f"{Fa2}/{C_0_podsh_t} = {Fa2/C_0_podsh_t}")
```

$$396.0/18100 = 0.021878453038674032$$

Определяю  $e$  по таблице 5.1

Для соотношения  $F_a/C_0 = 0,021$  и для  $\alpha = 12$  выбираю примерное  $e = 0,32$  (радиально-упорный шариковый однорядный)

Для

$$F_a/C_a = 0.021 < e = 0.32$$

Устанавливаю:

$$X = 1$$

$$Y = 0$$

Определяю соотношение  $F_a/(V F_r)$  для обоих подшипников

```
In [99]: V = 1
# Подшипник A
print(f"{Fa2}/({V}*{Ra_sum_2}) = {Fa2/(V*Ra_sum_2)}")

# Подшипник B
print(f"{Fa2}/({V}*{Rb_sum_2}) = {Fa2/(V*Rb_sum_2)}")
```

$396.0/(1*2612.415845436214) = 0.151583830228176$

$396.0/(1*2282.4446890973136) = 0.1734981802150984$

В обоих случаях соотношение  $F_a/(V F_r)$  меньше  $e$

```
In [100... # Устанавливаю e
e_podsh_t = 0.32
X_podsh_t = 1
Y_podsh_t = 0
```

```
In [101... def resurs2():

    # Частота оборотов
    n = n2

    # Подшипник: данные
    # 36207
    Cr = 24000
    C0 = C_0_podsh_t
    alpha = 12
    e = e_podsh_t
    X = X_podsh_t
    Y = Y_podsh_t

    V = 1
    Kb = 1.3
    Kt = 1
    p = 3

    # -----Силы-----
    #
    # (Fa) --->
```

```

# (A)---> <---(B)
#
# Силы от подшипников и осевая сила
# направлены как на рисунке: Vab == True
#
# Если сила Fa направлена в другую сторону
# Установить флаг Vab == false или указать
# силу со знаком минус

Vab = False

# Силы: указывать в Ньютонах
Fa = Fa2
Ra = Ra_sum_2
Rb = Rb_sum_2

print("Входные данные")
print("\nСилы:")
print(f"Fa = {Fa}")
print(f"Ra = {Ra}")
print(f"Rb = {Rb}")

print("\nПодшипник:")
print(f"Cr = {Cr}")
print(f"C0 = {C0}")
print(f"e = {e}")
print(f"X = {X}")
print(f"Y = {Y}")

print("\nПостоянные:")
print(f"p = {p}")
print(f"V = {V}")
print(f"Kb = {Kb}")
print(f"Kt = {Kt}")

if Fa < 0:
    Vab = False
    Fa = -Fa

S1 = e * Ra
S2 = e * Rb

if Vab:
    F1 = S1 if (-S1 - Fa + S2) < 0 else S2 - Fa
    F2 = S2 if (S1 + Fa - S2) < 0 else S1 + Fa
else:
    F1 = S1 if (-S1 + Fa + S2) < 0 else S2 + Fa
    F2 = S2 if (S1 - Fa - S2) < 0 else S1 - Fa

f_ek = lambda r, a: (X*V*r + Y*a)*Kb*Kt

print(f"Fa(A)/C0 = {F1/C0:.1f}")
print(f"Fa(B)/C0 = {F2/C0:.1f}")

```



```

F1_ek = f_ek(Ra, F1)
F2_ek = f_ek(Rb, F2)

print("\nВыходные данные")
print(f"Fa = {Fa}")
print(f"Ra = {Ra:.1f}")
print(f"Rb = {Rb:.1f}")
print(f"S1 = {S1:.1f}")
print(f"S2 = {S2:.1f}")
print(f"F1 = {F1:.1f}")
print(f"F2 = {F2:.1f}")
print(f"Fa1/VFr1 = {F1/Ra:.1f} > 0.35 - нужен радиально-упорный подшипник")
print(f"Fa2/VFr2 = {F2/Rb:.1f} > 0.35 - нужен радиально-упорный подшипник")
print(f"F1_ek = {F1_ek:.1f}")
print(f"F2_ek = {F2_ek:.1f}")

f_l = lambda x: (Cr/x) ** p
L1 = f_l(F1_ek)
L2 = f_l(F2_ek)

print(f"Срок службы 1-го подшипника = {L1:.0f} млн.об")
print(f"Срок службы 2-го подшипника = {L2:.0f} млн.об")

f_lh = lambda x: 1e6 * x / (60 * n)
lh1 = f_lh(L1)
lh2 = f_lh(L2)

print(f"Срок службы 1-го подшипника = {lh1:.0f} часов")
print(f"Срок службы 2-го подшипника = {lh2:.0f} часов")

resurs2()

```

## Входные данные

Силы:

$F_a = 396.0$

$R_a = 2612.415845436214$

$R_b = 2282.4446890973136$

Подшипник:

$C_r = 24000$

$C_0 = 18100$

$e = 0.32$

$X = 1$

$Y = 0$

Постоянные:

$p = 3$

$V = 1$

$K_b = 1.3$

$K_t = 1$

$F_a(A)/C_0 = 0.1$

$F_a(B)/C_0 = 0.0$

## Выходные данные

$F_a = 396.0$

$R_a = 2612.4$

$R_b = 2282.4$

$S_1 = 836.0$

$S_2 = 730.4$

$F_1 = 1126.4$

$F_2 = 730.4$

$F_{a1}/V_{Fr1} = 0.4 > 0.35$  - нужен радиально-упорный подшипник

$F_{a2}/V_{Fr2} = 0.3 > 0.35$  - нужен радиально-упорный подшипник

$F_{1\_ek} = 3396.1$

$F_{2\_ek} = 2967.2$

Срок службы 1-го подшипника = 353 млн.об

Срок службы 2-го подшипника = 529 млн.об

Срок службы 1-го подшипника = 104069 часов

Срок службы 2-го подшипника = 156045 часов

---

## Конструирование корпуса

Основные элементы корпуса из чугуна (табл. 6.1 стр 32)

```
In [102... # Толщина стенки корпуса редуктора
po_stenka = 0.04*a_w + 2
print(po_stenka)
```

7.0

```
In [103... # Толщина крышки редуктора
po_1_krishka = 0.032*a_w + 2
print(po_1_krishka)
```

6.0

```
In [104... # Толщина верхнего пояса(фланца) корпуса
b_korp = 1.5*po_stenka
print(b_korp)
b_korp = 10
print(b_korp)
```

10.5  
10

```
In [105... # Толщина нижнего пояса (фланца) крышки корпуса
b_krishka = 1.5*po_l_krishka
print(b_krishka)
```

9.0

```
In [106... # Толщина нижнего пояса корпуса
# При наличии бобышки
p1_korp = 1.5*po_stenka
print(p1_korp)
p1_korp = 10
print(p1_korp)

p2_korp = 2.25 * po_stenka
print(p2_korp)
p2_korp = 16
print(p2_korp)
```

10.5  
10  
15.75  
16

```
In [107... # Толщина ребер основания корпуса
m_korp = 0.85*po_stenka
print(m_korp)
m_korp = 6
print(m_korp)
```

5.95  
6

```
In [108... # Толщина ребер крышки
m1_krishka = 0.85*po_l_krishka
print(m)
m1_krishka = 5
print(m)
```

4.0  
4.0

```
In [109... # Диаметр фундаментных болтов (их число больше либо равно 4) (округляют до c
d1_bolt = 0.03 * a_w + 12
print(d1_bolt)
d1_bolt = 16
print(d1_bolt)
```

15.75  
16

```
In [124... # Диаметр болтов у подшипников
d2_bolt = 0.7*d1_bolt
print(d2_bolt)
d2_bolt = 12
print(d2_bolt)
```

11.2  
12

```
In [111... # Диаметр болтов соединяющих основание корпуса с крышкой
d3_bolt = 0.5*d1_bolt
print(d3_bolt)
```

8.0

```
In [112... # Размеры определяющие положение болтов d2
e_bolt = 1*d2_bolt
print(e_bolt)
```

12

```
In [113... # Размеры определяющие положение болтов d2

d4_bolt = 8 # Диаметр винтов крепления крышки подшипников (табл. 6.2)
q_bolt = 0.5 * d2_bolt + d4_bolt
print(q_bolt)
```

14.0

Дополнительные элементы корпуса из чугуна (табл. 6.1 стр 32)

Гнездо под подшипник

```
In [114... # Винты крепления крышки подшипника d4
print(d4_bolt)
```

8

```
In [115... # Число винтов
n_vint = 4
print(n_vint)
```

4

```
In [116... # Диаметр гнезда для торцевых крышек подшипника
# Наружный диаметр подшипника 7606
D_t = 72
# Диаметр с припуском
Dk_t_prip = D_t + 2
print(Dk_t_prip)
```

74

```
In [117... # Диаметр гнезда для торцевых крышек подшипника
# Наружный диаметр подшипника 36207
D_b = 72
```

```
# Диаметр с припуском (торцевая крышка)
Dk_b_prip = D_b + 2
print(Dk_b_prip)
```

74

```
In [118... # Длина гнезда минимальная
# l_gnez = B(Ширина подшипника) + K2(высота буртика удерживающего подшипник)
# Ширина подшипника 36207
B_podsh = 17
l_gnez_t = B_podsh + 4 + 4
print(l_gnez_t)
```

25

```
In [119... # Длина гнезда минимальная
# l_gnez = B(Ширина подшипника) + K2(высота буртика удерживающего подшипник)
# Ширина подшипника 7606
B_podsh_b = 29
l_gnez_b = B_podsh_b + 4 + 5
print(l_gnez_b)
```

38

## Размеры штифта

```
In [120... # Диаметр
d_shtift = d3_bolt
print(d_shtift)
```

8.0

```
In [121... # Длина
l_shtift = b_korp + b_krishka + 5
print(l_shtift)
```

24.0

## Наименьший зазор между наружной поверхностью колеса и стенкой корпуса

```
In [122... # По диаметру и по торцам
x_zazor = 1.1 * po_stenka
print(x_zazor)
x_zazor = 8
print(x_zazor)
```

7.7000000000000001

8

```
In [125... # До днища от колеса
Y_zazor = 40
print(Y_zazor)
```

40

In [ ]: