Group Assignment 1

ME 321 - Analysis for Design

Cameron Galibois, Jay Parmar, Chandler Wimmer

Fluid Power, Mechanical Power, and Forces/Moment Calculations

The fluid and mechanical powers can be calculated using the formulas below and values provided. The forces and moments from the pump weight and the pulley tension can also be found using the work below.

1.) Required fluid power:

$$\frac{Q \times H \times SG}{3960} = \frac{150 \times 60 \times 1}{3690} = \boxed{2.27\ HP} \quad \text{or } 1695W$$

2) Mechanical Shaft Power

$$\frac{P_{fluid,hp}}{\eta} = \frac{2.27}{0.65} = \boxed{3.50\,HP} \quad \text{or } 2610W$$

3.) Forces/Moment on Platform (pump weight, belt load, rxns)

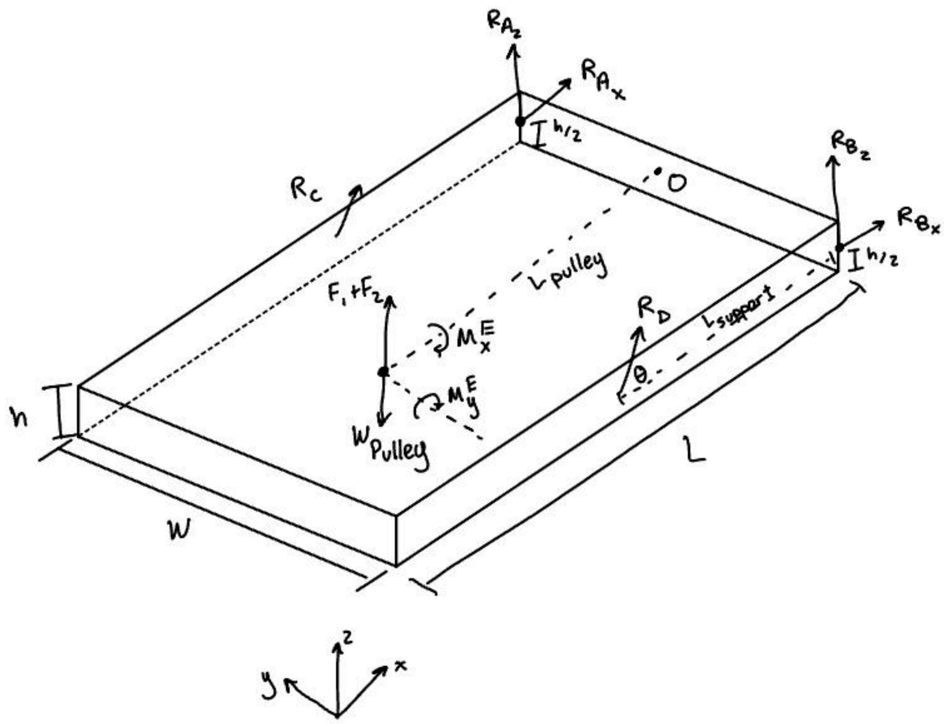
$$P_{HP} = \frac{T \times N}{5252} \rightarrow T = \frac{5252\,P_{HP}}{N} = \frac{5252\,(3.5)}{1800} = \boxed{10.2\ lb \cdot ft} \text{ or } 13.8\,Nm \qquad T_{tight} = 200 + \frac{13.8}{0.127} = \boxed{309N}$$

$$T_{tight} = 200 + 309 = 509N = \boxed{114.5\ lbf}$$

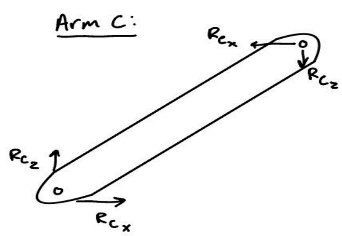
$$F_{net,pump} = 114.5 - 12.5 = \boxed{102\ lbf}$$

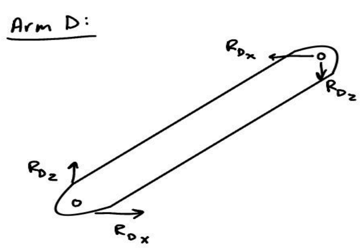
Platform Free Body Diagram

The platform will experience a group of forces and moments from the supports and pulley. The reactions from the supports will have both x and z components. The arms supporting the platform will also be applying a force to the wall that will be equal and opposite in direction to the force applied on the platform.



Support Arm Free Body Diagram





Code for Forces/Moments, Load on Each Support Arm & Connection Points

The code below was used to calculate the powers, forces, and moments necessary to analyze the system. Some constants were used to solve for these values based on the dimensions provided. Once the design is finalized, these constants can be changed to provide a more accurate set of forces.

```
import math
import numpy as np
from sympy import symbols, Eq, solve

def fluid_power(Q_ft, rho, g, h, n_p):
   p_fluid = (rho * g * Q_ft * h) / 550
   p_mechanical = p_fluid / n_p
   return p_fluid, p_mechanical

def tensions(p_mechanical, RPM, Pulley_rad, F1):
   F2 = ((p_mechanical * 550) / ((Pulley_rad / 12) * (RPM * 2 * np.pi / 60))) +
F1
   return F2

def moments(F1, F2, L_pulley, L_pump):
   MEy = (F1 + F2) * (L_pulley - L_pump)
   MEx = (F2 - F1) * (Pulley_rad)
   return MEy, MEx

def forces(F1, F2, W_beam, W_pump, theta, L_pump, L_support, L_beam,
beam_width, MEy, MEx):
   Rc, Rd, Wall_X, Wall_Z = symbols('Rc Rd Wall_X Wall_Z')
   F_z = Eq(F1 + F2 - W_pump - Rc * math.sin(theta) - Rd * math.sin(theta) -
W_beam + Wall_Z, 0)
   M_o_x = Eq(MEx - ((Rc*np.sin(theta) + (Wall_Z/2)) * (beam_width/2)) +
((Rd*np.sin(theta) + (Wall_Z/2)) * (beam_width/2)), 0)
   F_x = Eq(Rc * math.cos(theta) + Rd * math.cos(theta) + Wall_X, 0)
   M_o_y = Eq(MEy + ((F1+F2-W_pump) * L_pump) - ((Rc * math.sin(theta) + Rd *
math.sin(theta)) * L_support) - (W_beam * (L_beam/2)), 0)

   print("Symbolic Equations:")
   print(f"Sum of forces in the z-direction: {F_z}")
   print(f"Sum of forces in the x-direction: {F_x}")
   print(f"Moment at the origin about the y-axis: {M_o_y}")
   print(f"Moment at the origin about the x-axis: {M_o_x}")

   soln = solve((F_z, F_x, M_o_y, M_o_x), (Rc, Rd, Wall_X, Wall_Z))
```

```python
    print("Numeric Solutions:")
    print(f"Rc = {soln[symbols('Rc')]}")
    print(f"Rd = {soln[symbols('Rd')]}")
    print(f"Wall_X = {soln[symbols('Wall_X')]}")
    print(f"Wall_Z = {soln[symbols('Wall_Z')]}")
    return soln

def components(soln, theta):
    Rc_X = -soln[symbols('Rc')] * math.cos(theta)
    Rc_Z = -soln[symbols('Rc')] * math.sin(theta)
    Rd_X = -soln[symbols('Rd')] * math.cos(theta)
    Rd_Z = -soln[symbols('Rd')] * math.sin(theta)
    return Rc_X, Rc_Z, Rd_X, Rd_Z

# Variables to change
L_mounting = 6 # Vertical distance between mounting locations [in]
L_support = 6 # Horizontal placement [in]
L_beam = 18 # Horizontal placement [in]
Thick_beam = 0.5 # [in]
beam_width = 8 # [in]
density_beam = 0.1 # [lb/in^3]
```

```python
# Values
Q = 150 # Volume flow rate [GPM]
Q_ft = Q / 7.48 / 60 # Volume flow rate [ft^3/sec]
n_p = 0.65 # Pump Efficiency
h = 60 # Height [ft]
g = 32.174 # Acceleration Due to Gravity
W_pump = 12.5 # Weight of Pump [lbf]
rho = 1.94 # Density [slugs/ft3]
RPM = 1800 # RPM of Motor
L_pump = 9.56 + 2.5 # Distance from pump to wall [in]
L_pulley = 18.56 # Distance from pulley to wall [in]

# Calculated Values
W_beam = density_beam * L_beam * beam_width * Thick_beam # Weight of Beam [lbf]
theta = np.arctan(L_mounting / L_support)

p_fluid, p_mechanical = fluid_power(Q_ft, rho, g, h, n_p)
print(f"Fluid Power: {p_fluid:.6f} hp \nMechanical Power: {p_mechanical:.6f} hp")

F1 = 44.9618 # Pulley Tension 1 [lbf]
```

```python
Pulley_rad = 5 # [in]
F2 = tensions(p_mechanical, RPM, Pulley_rad, F1)
print(f"Pulley Tension 1: {F1:.6f} lbf")
print(f"Pulley Tension 2: {F2:.6f} lbf")

MEy, MEx = moments(F1, F2, L_pulley, L_pump)
print(f"Moment at Point E: {MEy:.6f} lbf-in")
print(f"Torque at Point E: {MEx:.6f} lbf-in")

soln = forces(F1, F2, W_beam, W_pump, theta, L_pump, L_support, L_beam,
beam_width, MEy, MEx)
print(f"Force Rc: {soln[symbols('Rc')]:.6f} lbf")
print(f"Force Rd: {soln[symbols('Rd')]:.6f} lbf")

Rc_X, Rc_Z, Rd_X, Rd_Z = components(soln, theta)
print(f"Force Rc Upper X-Component: {Rc_X:.6f} lbf")
print(f"Force Rc Upper Z-Component: {Rc_Z:.6f} lbf")
print(f"Force Rd Upper X-Component: {Rd_X:.6f} lbf")
print(f"Force Rd Upper Z-Component: {Rd_Z:.6f} lbf")

print(f"Force Wall_X: {soln[symbols('Wall_X')]:.6f} lbf")
print(f"Force Wall_Z: {soln[symbols('Wall_Z')]:.6f} lbf")
```

Code Output:

```
Fluid Power: 2.275798 hp
Mechanical Power: 3.501228 hp
Pulley Tension 1: 44.961800 lbf
Pulley Tension 2: 69.480261 lbf
Moment at Point E: 743.873398 lbf-in
Torque at Point E: 122.592306 lbf-in
Symbolic Equations:
Sum of forces in the z-direction: Eq(-0.707106781186547*Rc -
0.707106781186547*Rd + Wall_Z + 94.7420612334059, 0)
Sum of forces in the x-direction: Eq(0.707106781186548*Rc +
0.707106781186548*Rd + Wall_X, 0)
Moment at the origin about the y-axis: Eq(-4.24264068711928*Rc -
4.24264068711928*Rd + 1908.49465649201, 0)
Moment at the origin about the x-axis: Eq(-2.82842712474619*Rc +
2.82842712474619*Rd + 122.59230616703, 0)
Numeric Solutions:
Rc = 246.589714996967
Rd = 203.246789490964
Wall_X = -318.082442748669
```

```
Wall_Z = 223.340381515263
Force Rc: 246.589715 lbf
Force Rd: 203.246789 lbf
Force Rc Upper X-Component: -174.365260 lbf
Force Rc Upper Z-Component: -174.365260 lbf
Force Rd Upper X-Component: -143.717183 lbf
Force Rd Upper Z-Component: -143.717183 lbf
Force Wall_X: -318.082443 lbf
Force Wall_Z: 223.340382 lbf
```

The connection forces between the arms and the wall at the top of the arms are labeled as Rc Upper x-component, Rc Upper Z-component, Rd Upper x-component, and Rd Upper Z-component. The connection forces for the wall are labeled as Wall_X and Wall_Z.

V-M Diagram:

Theoretical V-M diagrams are graphed below using python's matplotlib feature. While there are no explicit values stated, we can assume that the maximum bending moment is located near the support arms. The shear starts based on the forces with the pulley system. The large change in shear is due to the force applied by the support arms.