

Lab_report_TEAM12

October 8, 2025

1 Assignment

In this assignment, we will design the drive system for a transport belt. The drive will be powered by a geared electromotor and a chain transmission. Here are the initial specifications:###

Speed of the motor: $n_3 = 1500 \text{ rpm}$

Team specific requirements: - Speed of chain: $v_1 = 15 \text{ km/h}$ - Sprocket diameter: $d_1 = 450 \text{ mm}$ - Power needed to move the carts: $P_1 = 4.4 \text{ kW}$ —

```
[167]: # import statements
import sympy as sp
import numpy as np
import matplotlib.pyplot as plt

import rm_python_lib.MechDesign.Helpers as HM

from rm_python_lib.MechDesign.Units.Units import m_, mm_, kg_, s_, N_, rpm_, W_
import rm_python_lib.MechDesign.Units.UnitMethods as UM

import rm_python_lib.MechDesign.RnM as RnM
belt = RnM.Belt()
gears = RnM.GearDesign()
chain = RnM.Chain()
B1 = RnM.Shaft()
B2 = RnM.Shaft()
connections = RnM.ShaftConnection()
```

1.1 Sesion 1

1.1.1 1. P T n Calculations

First let's insert known data to the variables. The belt efficiency is taken from the chapter 16 from Roloff and Matek Book that explains the flat belts (the ones we will be using). Gear efficiency is assumed based on the conditions in Part 3 of this assignment.

```
[168]: # Given values
belt.n_1 = 1500*rpm_ # Speed of motor
chain.v = 15/3.6*m_/s_ # Speed of sprocket 1
```

```

chain.d_1 = UM.All_to_SI(450*mm_) # Diameter of sprocket 1
chain.P_2 = 4.4*1000*W_ # Power needed at sprocket 1
gears.beta_1 = 0 # teeth are straight cut
gears.beta_2 = 0 # teeth are straight cut
gears.nu_t = 0.9 # gear transmission efficiency
belt.eta = 0.98 # belt drive efficiency

```

Having in mind the given data now it is possible to calculate the parameters for the shaft $B2$. Also it is important to mention that the shafts are numbered from left to right. Meaning that motor shaft data is marked with a “ $_1$ ”, $B1$ shaft with and index “ $_2$ ” and the $B2$ shaft with and index “ $_3$ ”. Since real operating conditions include transmission losses, the power decreases along the system, and each successive shaft transmits slightly less power.

```

[169]: chain.T_1 = chain.P_2/(2*chain.v/chain.d_1) # Torque on sprocket 1

w_1 = 2*chain.v/chain.d_1 # Angular velocity of sprocket 1
chain.n_1 = UM.RemoveUnits(w_1/(2*sp.pi)*60*rpm_)*rpm_ # Rotational speed of
↳ sprocket 1

gears.P_1 = chain.P_2/ gears.nu_t # Power needed at gear 1
belt.P = gears.P_1 / belt.eta # Power needed at pulley 1

t=HM.EqPrint('P_1', belt.P)
t=HM.EqPrint('P_2', gears.P_1)
t=HM.EqPrint('P_3', chain.P_2)
t=HM.EqPrint('T_3', chain.T_1.evalf())
t=HM.EqPrint('n_3', chain.n_1.evalf())

```

$$P_1 = 4989.0W$$

$$P_2 = 4889.0W$$

$$P_3 = 4400.0W$$

$$T_3 = 237.6Ws$$

$$n_3 = 176.8rpm$$

```

[170]: import math
i_tot = chain.n_1/belt.n_1
t=HM.EqPrint('i_tot',i_tot)

belt.n_2 = math.sqrt(UM.RemoveUnits(3*chain.n_1*belt.n_1/5))
t = HM.EqPrint('n_2',belt.n_2*rpm_)

belt.i = UM.RemoveUnits(belt.n_2/belt.n_1)
t = HM.EqPrint('i_r',belt.i)

```

```
gears.omega_1 = UM.rpm_to_rad_s(belt.n_2*rpm_)
gears.omega_2 = UM.rpm_to_rad_s(chain.n_1)

gears.i = gears.omega_2/gears.omega_1
t = HM.EqPrint('i_twk', gears.i)
```

$$i_{tot} = 0.1179$$

$$n_2 = 398.9rpm$$

$$i_r = 0.266$$

$$i_{twk} = 0.4433$$

Since now all the rotational speeds are known the torques can be computed.

```
[171]: belt.T_1 = belt.P/ UM.rpm_to_rad_s(belt.n_1)

belt.T_2 = gears.P_1/ gears.omega_1
t = HM.EqPrint('T_1', belt.T_1)
t = HM.EqPrint('T_2', belt.T_2)
```

$$T_1 = 31.76Ws$$

$$T_2 = 117.0Ws$$

1.2 ### 2. Motor selection

1.2.1 Selection Criteria

Based on the provided data of desired Power output = 4.4kW, and with the belt efficiency η (= 0.98), and the gear efficiency η_g (assumption $\eta_g = 0.9$), and $k_A = 1.9$, we can find that the motor desired power is around 9.48kW. Looking at the “Power P in kW for a synchronous rotational speed, 1500 rpm-1 column, we notice there is no 9.48kW value. Therefore we need to take the upper suitable value, 11kW. And hence the selected motor is 160M. It has a shaft diameter of 42mm (needed for the key calculation).

Table 16-21 Surface cooled induction motors with squirrel-cage rotor according to DIN 42 673 T1

DIN 42673 T1

DIN 2211 T3

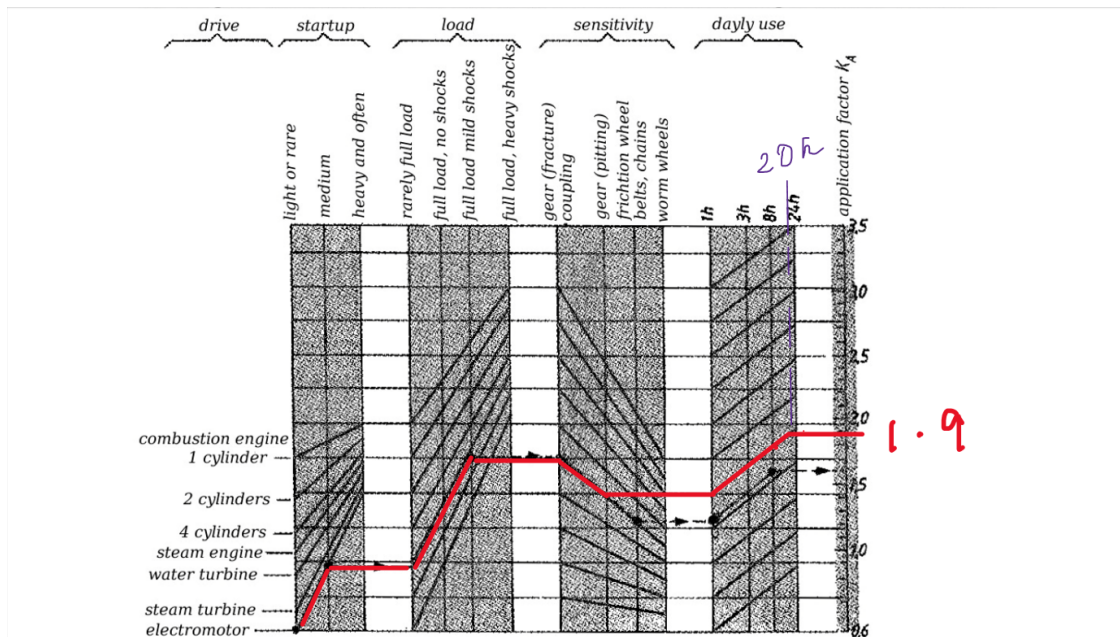
acceptable axle load

dimensions in mm

size	h	a	b	w ₁	s	axle ending d_2 in mm at n_s in min ⁻¹		power P in kW for a synchronous rotational speed in min ⁻¹ 2)				inertia of the rotor J in kgm ² 4)		T ₀ /T _{0.95} 4) 5)		acceptable axle load for $n_s = 1500$ min ⁻¹ , for a force at:		narrow profile pulley DIN 2211 7) at $n_s = 1500$ min ⁻¹		coupling at $n_s = 1500$ min ⁻¹ e.g. type	
						3000	≤ 1500	3000	1500	1000	750	3000	1500	3000	1500	$x=0$ F ₀ in kN	$x=1$ F ₁ in kN	profile	d _f mm	grooves z	Periflex size
71	71	90	112	45	M6	14 × 30	0.37	0.25	—	—	0.00034	0.00054	2.0	1.9	0.45	0.53	—	—	—	1	24
71	71	90	112	45	M6	14 × 30	0.55	0.37	0.18	—	0.00039	0.00068	2.4	2.25	0.45	0.53	—	—	—	1	24
80	80	100	125	50	M8	19 × 40	0.75	0.55	0.37	—	0.00089	0.00134	2.2	2.0	0.52	0.63	SPZ	63	1	1	24
80	80	100	125	50	M8	19 × 40	1.1	0.75	0.55	—	0.00120	0.00182	2.4	2.3	0.52	0.63	SPZ	63	1	1	24
90S	90	100	140	56	M8	24 × 50	1.5	1.1	0.75	—	0.00210	0.00316	2.6	2.2	0.78	0.92	SPZ	71	1	2	24
90L	90	125	140	56	M8	24 × 50	2.2	1.5	1.1	—	0.00250	0.00383	3.0	2.6	0.78	0.92	SPZ	71	2	2	24
100L	100	140	160	63	M10	28 × 60	3	2.2	1.5	0.75 3)	0.00325	0.00488	2.8	2.4	1.06	1.31	SPZ	90	2	2	28
112M	112	140	190	70	M10	28 × 60	4	2.2	1.5	—	0.0055	0.0094	3.3	2.5	1.04	1.27	SPZ	112	2	2	28
132S	132	140	216	89	M10	38 × 80	5.5 3)	5.5	3	2.2	0.0080	0.0180	3.4	3.4	1.53	1.94	SPZ	125	2	6	38
132M	132	178	216	89	M10	38 × 80	—	7.5	4 3)	3	—	0.0318	—	3.1	1.53	1.94	SPZ	140	3	6	38
160M	160	210	254	108	M12	42 × 110	11 3)	11	7.5	4 3)	0.0230	0.045	4.0	3.6	1.59	2.04	SPZ	140	4	16	42
160L	160	254	254	108	M12	42 × 110	18.5	15	11	7.5	0.0615	0.101	2.9	2.7	1.59	2.04	SPZ	140	5	16	48
180M	180	241	279	121	M12	48 × 110	22	18.5	—	—	0.0753	0.118	3.0	2.9	1.95	2.35	SPZ	160	5	16	48
180L	180	279	279	121	M12	48 × 110	—	22	15	11	—	0.141	—	3.0	1.95	2.35	SPA	180	4	16	55
200L	200	305	318	133	M16	55 × 110	30 3)	30	18.5 3)	—	0.142	0.222	2.6	2.7	2.75	3.35	SPA	180	5	40	55
225S	225	286	356	149	M16	55 × 110 60 × 140	—	37	—	18.5	—	0.356	—	2.8	2.95	3.75	SPA	200	5	40	60
225M	225	311	356	149	M16	55 × 110 60 × 140	45	45	30	22	0.270	0.461	3.0	2.8	2.95	3.75	SPB	224	4	40	65
250M	250	349	406	168	M20	60 × 140 65 × 140	55	55	37	30	0.424	0.677	2.7	2.7	3.60	4.40	SPB	224	4	40	75
280S	280	368	457	190	M20	65 × 140 75 × 140	75	75	45	37	0.816	1.06	3.2	2.7	7.20	8.70	—	—	—	63	75
280M	280	419	457	190	M20	65 × 140 75 × 140	90	90	55	45	0.957	1.26	3.2	2.7	7.20	8.70	—	—	—	63	85
315S	315	406	508	216	M24	65 × 140 80 × 170	110	110	75	55	1.19	2.00	3.2	2.8	8.10	9.90	—	—	—	125	85
315M	315	457	508	216	M24	65 × 140 80 × 170	132	132	90	75	1.45	2.35	3.2	2.8	8.10	9.90	—	—	—	125	100

1.3 ### 3. Finding the right K_A and efficiency

In this section, K_A and the efficiency of the motor is calculated based on some design assumptions.



The operating conditions correspond to medium-duty service, characterized by a moderate number of startups, full-load operation with mild shocks, and a drivetrain susceptible to gear pitting. Based on these assumptions and an average daily runtime of 20 hours, the estimated operating factor is $K_A = 1.9$. This factor accounts for dynamic effects such as torque fluctuations and transient load peaks, representing the ratio between equivalent dynamic torque and nominal steady torque. Consequently, the effective efficiency of the motor–gear system will be slightly lower than the rated

efficiency due to mechanical losses and vibration-induced stresses. This estimation serves as a preliminary value and should be refined through vibration analysis, load monitoring, and thermal efficiency testing under real operating conditions.

1.3.1 4. Key calculations

The dimensions and type of the key are constrained by the shaft's characteristic. For general form fitting with high torque capacity, the DIN 6885 standard is suitable due to its large contact surface. Another significant reason is that this standard favors no axial movement, which is the case here.

1. Rationale for Key Type A (Rounded Ends) We have chosen Key Type A (with rounded ends) over Type B (with squared ends) because of ease on construction and higher availability in course materials.
2. Based on the tables available in “2 - Table connections” file, and the motor selection results above, the dimensions of our keys and shafts are:
 - Shaft diameter: 42mm
 - Key width b: 12mm
 - key height h: 8mm
 - Depth key way (shaft) t1: 5mm
 - Depth key (hub) t2: 3.3mm
 - Standard length l (max): 110mm (we pick the min value between motor's length(110mm) and key's length (140mm))
3. Based on Table 12-1a, we can only record constraints of the hub's dimension. Assume that our hub is made out of steel, its diameter varies from 75.6 to 84mm, while its length is between 46.2 and 58.8mm.
4. We then use the existing formula at Formula 12.1 or the built-in function to calculate the key's surface pressure on the side at its weakest point and compare it with yield strength value (p_{hat}) obtained from the materials. The material's characteristics can be obtained from the file “General_Tables_RM_material_properties_EN”.
 - Shaft/Hub: yield strength $\text{ReN} = 550\text{N/mm}^2$ (Material 38Cr2 - from assignment description)
 - Key: yield strength $\text{ReN} = 295\text{N/mm}^2$ (Material E295 - From Chapter 12)

There are two ways of finding surface pressure and the threshold p_{hat} value: method B and C. In average surface pressure calculation, method B/C influences the value of K_{lamda} : $K_{\text{lamda}} = 1$ for C and according to figure 12-2c (with reference based on $l'/d = (l-b)/d$). In threshold calculation, method B/C influences the formula more. Here, because of our material's characteristic, the formula for ductile material is used. In this assignment, method C is chosen because of the dominance of uneven surface loading factor due to relative rotation of shaft/hub to alternating torques in method B.

```
[172]: # creating a new shaft connection, (this is 1 specific connection in your
      ↪ design)
      SC = RnM.ShaftConnection()
```

```

# setting up constants, based on assignment (units are specified using a
↳trailing '_')
SC.d = 42*mm_
SC.T_nom = 70.0282*N*_m_
SC.K_A = 1.9

# setting up constants that are not likely to change
SC.phi = 1 # 1key
SC.n = 1 # 1 key
SC.K_lambda = 1 # method C

SC.K_t = 0.97
SC.b = 12*mm_
SC.h = 8*mm_

# setting up constants chosen more arbitrary
# guessing a first value for key length
SC.l = 63*mm_ # when using l'<=1.3*d with a shaft of 60mm and key width of 18
↳mm l'<= 78 and l<78+18=96mm and a DIN 116 A60 has max length of 85mm

# setting up helper functions
SC.lprime = SC.E12_1_hI_KeyEffectiveLength()
SC.hprime = SC.E12_1_hJ_KeyEffectiveHeight()
SC.T_eq = SC.E12_1_hC_DynamicLoadTorque()

# setting up the main equations
SC.p_gem = SC.E12_1B_KeyAveragePressure()
HM.EqPrint("p_gem",SC.p_gem)
# convert units, the previous expression still has both m_ and mm_
subst = {m_:1e3*mm_}
SC.p_gem = SC.p_gem.evalf(subs=subst) # this line will substitute the symbol
↳'m_' with '1e3*mm_'
t=HM.EqPrint('p_gem',SC.p_gem)

```

$$p_{gem} = \frac{0.03451Nm}{mm^3}$$

$$p_{gem} = \frac{34.51N}{mm^2}$$

```

[173]: # The numerator constant C_num = 2 * T_eq * K_lambda (in N.mm)
C_num_sym = 2 * SC.T_eq * SC.K_lambda
C_num_val = UM.RemoveUnits(C_num_sym.evalf(subs=subst))

# The denominator constant C_denom = d * h' (in mm^2)
C_denom_val = UM.RemoveUnits((SC.d * SC.hprime*SC.n*SC.phi))

# Final Plot Constant C_plot = C_num / C_denom (in N/mm^2 * mm)

```

```

C_plot = C_num_val / C_denom_val
# C_plot is approx. 2100 N/mm2 * mm
HM.EqPrint('C_plot', C_plot)
HM.EqPrint('C_denom_val', SC.hprime)

```

$$C_{plot} = 1760.0$$

$$C_{denomval} = 3.6mm$$

[173]: $C_{denomval} = 3.6mm$

```

[174]: # getting relevant limits for the surface pressure
R_eNShaft = 550* N_/mm**2
K_tShaft = 0.97
R_eNKey = 295* N_/mm**2
K_tKey = K_tShaft
R_mNGear = 1000* N_/mm**2
K_tGear = 1
# calculate the reference strengths
R_eShaft = R_eNShaft * K_tShaft
R_eKey = R_eNKey * K_tKey
R_mGear = R_mNGear * K_tGear

# display the results
HM.EqPrint('R_eShaft',R_eShaft)
HM.EqPrint('R_eKey',R_eKey)
HM.EqPrint('R_mHub',R_mGear)

S_f = 1.5 #Assumption

# Corrected Yield Strengths (R_e = R_eN * Kt)
R_e_Shaft_corr = R_eNShaft * K_tShaft
R_e_Key_corr = R_eNKey * K_tKey

# --- FIX: REMOVE UNITS FROM P_HAT FOR PLOTTING ---
p_hat_Shaft = UM.RemoveUnits(R_e_Shaft_corr) / S_f
p_hat_Key = UM.RemoveUnits(R_e_Key_corr) / S_f # TRUE WEAKEST LINK LIMIT

# calculate overall safety (Using the weakest link: R_eKey)
# Note: The safety check must use the weakest material (R_eKey)
p_applied_at_l = UM.RemoveUnits(SC.p_gem) # The applied pressure calculated_
↳using l=63mm

S = min(UM.RemoveUnits(R_eShaft),UM.RemoveUnits(R_eKey),UM.
↳RemoveUnits(R_mGear))*N_/mm**2 / SC.p_gem

```

```
t=HM.EqPrint('S',S)
```

$$R_{eShaft} = \frac{533.5N}{mm^2}$$

$$R_{eKey} = \frac{286.1N}{mm^2}$$

$$R_{mHub} = \frac{1000N}{mm^2}$$

$$S = 8.292$$

```
[175]: #Define the pressure function p_gem(l') for plotting (p(l') = C_plot / l')
def p_gem_function_of_l_prime(l_prime):
    """Calculates applied pressure (N/mm^2) for a given key length (mm)."""
    #l_prime[l_prime <= 1e-6] = 1e-6
    return C_plot / l_prime

# --- L to L' Conversion Constants ---
B_WIDTH = UM.RemoveUnits(SC.b) # Key width b = 12 mm
L_EFF_LIMIT_D = 1.3 * UM.RemoveUnits(SC.d) # l'_limit = 1.3 * 42 mm = 54.6 mm

# Calculate Minimum Required Length (l'_min)
l_prime_min = C_plot / p_hat_Key
l_min_total = l_prime_min + B_WIDTH # Total minimum length required for safety
HM.EqPrint('l_prime_min', l_prime_min)
HM.EqPrint('l_prime_total', l_min_total)
# Define Plot Range (L_prime_values runs from l'_min to l'_max)
L_prime_plot_range = np.linspace(28, 110, 400)
Pressure_values = p_gem_function_of_l_prime(L_prime_plot_range)

# CONVERSION STEP: Convert l' values to L values for the X-Axis
L_values_for_plot = L_prime_plot_range + B_WIDTH/1000

# --- Plot Generation ---
plt.figure(figsize=(10, 6))

# 1. Plot the Pressure Curve (p_gem) - X-Axis is L
plt.plot(L_values_for_plot, Pressure_values, label=f"Applied Pressure_␣
↪$p_{\{gem\}}(l')$ (Method B, $K_{\{\lambda\}}={SC.K_lambda}$)",␣
↪color='darkblue', linewidth=2)

# 2. Plot the TRUE System Limit (Weakest Link: E295 Key)
plt.axhline(y=p_hat_Key, color='red', linestyle='--', linewidth=2,
            label=f"Weakest Link Limit (E295 Key) $\overline{p}={p_hat_Key:.␣
↪1f}$ N/mm^2")

# 3. Plot the Shaft Limit (38Cr2) - Stronger Link for comparison
```



```

plt.axhline(y=p_hat_Shaft, color='red', linestyle=':', linewidth=1.0, alpha=0.6,
            label=f"Shaft Limit (38Cr2)  $\overline{p} = \{p\_hat\_Shaft:.1f\} \text{ N/mm}^2$ ")

# 4. SHADE REGION: Ineffective Length ( $l' > 1.3 * d$ )
# The shade starts at the total length  $L\_limit = l'\_limit + b$ 
L_limit_total = L_EFF_LIMIT_D + B_WIDTH # 54.6 mm + 12 mm = 66.6 mm
plt.axvspan(L_limit_total, 115, alpha=0.2, color='purple', hatch='/',
            label=f"Ineffective Length Region ( $l > 1.3d + b = \{L\_limit\_total:.1f\} \text{ mm}$ ")

# 5. Plot the Minimum Required Length ( $l\_min\_total$ )
plt.axvline(x=l_min_total, color='green', linestyle='-.',
            label=f"Required  $l_{\{min\}} = \{l\_min\_total:.2f\} \text{ mm}$ ")

# 6. Highlight the chosen length ( $l=63\text{mm}$ )
L_chosen_point = UM.RemoveUnits(SC.l)
l_prime_for_chosen_l = L_chosen_point - B_WIDTH
p_at_chosen_l = p_gem_function_of_l_prime(np.array([l_prime_for_chosen_l]))[0]

plt.plot(L_chosen_point, p_at_chosen_l, 'ro', markersize=7,
        markeredgecolor='black',
        label=f"Chosen Design Point ( $l = \{L\_chosen\_point:.0f\} \text{ mm}$ ):  $\{p\_at\_chosen\_l:.2f\} \text{ N/mm}^2$ ", zorder=5)

# --- Final Styling ---
plt.ylim(0, 800)
plt.xlim(20, 115)
plt.title(f"Key Bearing Pressure vs. Total Key Length ( $d = \{SHAFT\_DIAMETER\_MM:.0f\} \text{ mm}$ )", fontsize=14)
plt.xlabel("Total Key Length  $l$  (mm)", fontsize=12) # Label now reflects 'l'
plt.ylabel("Average Pressure  $p$  (N/mm2)", fontsize=12)

# Add Annotation for Weakest Link
plt.annotate('Weakest Link (E295)', xy=(L_values_for_plot[-1], p_hat_Key + 30),
            xytext=(85, p_hat_Key + 100),
            arrowprops=dict(facecolor='black', shrink=0.05, width=1.5),
            fontsize=10)

plt.legend(fontsize=10)
plt.grid(True, linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()

print(f"\n--- Final Key Design Summary ---")

```

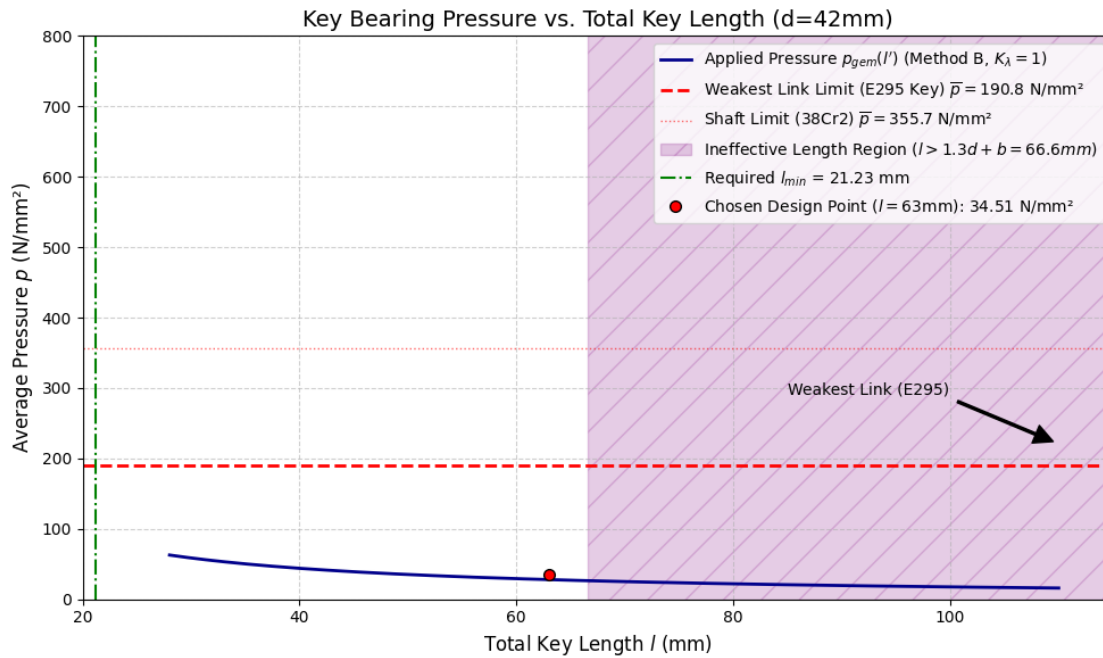
```

print(f"1. Maximum Applied Pressure (at L=28mm): {p_gem_function_of_l_prime(np.
    ↳array([28 - B_WIDTH]))[0]:.2f} N/mm^2")
print(f"2. Weakest Link Allowable Stress (p_hat): {p_hat_Key:.2f} N/mm^2")
print(f"3. Calculated Minimum Required Total Length (l_min): {l_min_total:.2f}↳
    ↳mm")
print(f"4. Chosen Standard Length (l): {L_chosen_point:.0f} mm (Safety Factor:↳
    ↳{p_hat_Key / p_at_chosen_l:.2f})")

```

$$l_{primemin} = 9.226$$

$$l_{primetotal} = 21.23$$



--- Final Key Design Summary ---

1. Maximum Applied Pressure (at L=28mm): 110.00 N/mm²
2. Weakest Link Allowable Stress (p_hat): 190.77 N/mm²
3. Calculated Minimum Required Total Length (l_min): 21.23 mm
4. Chosen Standard Length (l): 63 mm (Safety Factor: 5.53)

1.3.2 Conclusion:

From the code above for the calculations on key connections, the insight can be found:

- The p_{gem} pressure decreases as the key's length l increases.
- The lowest failure limit among all materials belongs to E295 for keys, which can be noticed.
- The maximum surface pressure that the key needs to withstand is well below the failure limit.