MOTOR AND MOTOR CONTROLLER COOLING SYSTEM DESIGN

BASIC HEAT LOSS CALCULATIONS

In the power train we use EMRAX 228 MV as our motor and BAMOCAR D3 inverter. These are the main power generating equipment in the power train. We referred the given data sheets and rule book to get initial understanding of the required system.

For the following calculations we have assumed that the all power loss is converted to heat.

For EMRAX 228 MV

It is stated in the data sheet that It has of 92% efficiency for the worst case. According to our design the maximum power output from the battery is 60kW. But this maximum value is not maintained throughout the lap. For a lap run we approximated the average power output from the battery to be 27kW using a simulation.

$$P_{l,motor} = 27 * (1 - 0.92)\ kW$$

For BAMOCAR D3

The data sheet doesn't specify any efficiency. But it states that the maximum power loss can be about 4kW in 700/400 configuration.

Table 01: Maximum power losses in the power train

Equipment	Maximum Power loss
EMRAX	2.16 kW
BAMOCAR D3	4 kW

So we have maximum of 6.16 kW power loss in our system. But designing cooling system capable of dissipation the maximum heat all the time will be a mistake since it makes the system much bigger. Hence for the design we considered the average power loss.

To get the average overall heat loss we assumed the tractive system efficiency to be 90% and continuous power of the vehicle as 27kW. So,

$$P_{avg} = 27 \times (1 - 0.9)$$
$$= 2.7\ kW$$

CHOICE OF COOLING

In January Kari Motor Speedway has max recorded temperatures of 33 degrees of Celsius. Because of the high air temperatures we have decided to go with liquid cooling instead of air cooling.

General design data related for the system was obtained from datasheets.

Table 02: Liquid cooling requirements for each component

Parameter	EMRAX 228	BAMOCAR D3
Minimum coolant flow rate	6 LPM	-
Maximum coolant flow rate	-	12LPM
Maximum coolant inlet temperature	50℃	65℃
Maximum inlet pressure	2 bar	6 bar
Maximum pressure drop across	0.3 bar	0.6 bar
Cut off temperature	90℃	80℃

COOLING LOAD CALCULATIONS

DESIGN PARAMETERS

In the calculations ambient temperature is taken to be 33℃ (recorded maximum ambient temperature at the Kari Motor Speedway in January). Then the radiator coolant temperature was taken as 45℃ to increase the temperature difference between air and the radiator (increased temperature difference causes increased heat transfer). And the initial coolant flow rate was taken as 12 LPM to achieve maximum possible heat transfer rate through the radiator.

Since we are first timers, obtaining radiator performance test data was not possible at this stage. The radiator makers does not publish their specifications on the internet.

We have decided to place our radiator in the side pod due to a couple of reasons

- Easier to do the ducting and achieve good air flow rate through the radiators
- Weight distribution of the vehicle will be reduced compared with the other options. Yaw inertia will be less.
- Easier to achieve lower center of gravity

At this stage we have basic sketch of our FSAE car. According to it we can accommodate 300mm x 200mm space in side pods for the radiators

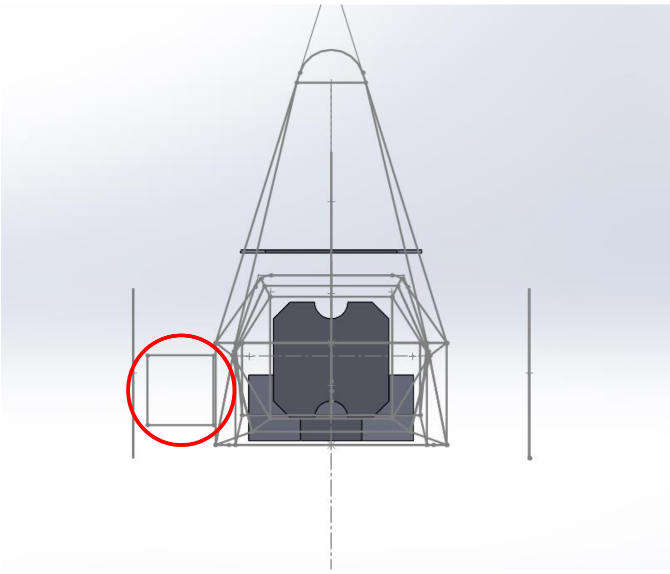

Figure 01: Estimated area for the radiator in one side pod (front view)

Market research was conducted to find a radiator which has the needed requirements. After that we came up with 2 options.

1. MISHIMOTO YFM700R – ATV radiator
2. MISHIMOTO DRZ400 (Left & Right) – Dirt bike radiators

MISHIMOTO was chosen due to their lifetime warranty and very positive internet reviews. And some of the radiator fin characteristics was available in their website.

For calculate the heat dissipation of a radiator with temperatures ε - NTU method can be used (since exit air temperature is unknown). But the biggest challenge is to find the overall heat transfer coefficient for the radiator without knowing any core characteristics of the radiator.

$$NTU = \frac{UA}{C_{\min}} \qquad \text{------- (1)}$$

The $C_{\min}$ can be calculated by knowing the mass flow rates for the air and coolant.

$$C_a = \dot{m}_a c_{p,a}$$
$$C_c = m_c \dot{c}_{p,a}$$

$$\dot{m}_c = coolant\ flow\ rate\left(\frac{kg}{s}\right)$$
$$m_a = air\ flow\ rate\ \left(\frac{kg}{s}\right)$$

The minimum of the C_a and C_c taken as $C_{\min}$ and maximum to be C_{max}

Then the heat capacity ratio,

$$C_r = \frac{C_{\min}}{C_{\max}}$$

Maximum possible heat dissipation can be calculated by using inlet air stream temperature of the both sides.

$$q_{\max} = C_{\min} \left(T_{c,in} - T_{a,in}\right)$$

Effectiveness (ϵ) can be calculated by using NTU and C_r values considering radiator as a crossflow heat exchanger.

$$\epsilon = 1 - e^{\left(\frac{1}{C_r}(NTU)^{0.28}\left(e^{-C_r(NTU)^{0.78}} - 1\right)\right)} \qquad \text{------- (2)}$$

Then the approximated heat dissipation can be calculated using

$$q = \epsilon q_{\max} \qquad \text{------- (3)}$$

According to the equation (1) q can be calculated if the UA is known for the radiator. It is the overall heat transfer rate per Kelvin of the radiator. This can be calculated according to [4] for a louvered fin radiator. In our calculations we have used [1] to get a better correlation for the Colburn – J factor.

To analyze both radiators in iterative manner, we made a python program. It will be available at our GITHUB

Code explanation

1) First we have defined a class called Radiator which we can use to generate radiator object by suppling radiator core characteristics. We couldn’t find all the radiator characteristics of our selected radiators. To find some parameters we basically used online images of the product. We approximated the louvered fin geometry. Radiator manufactures don’t publish these data.

```
class Radiator:

    def __init__(self, data):
        self.name = data['name']
        self.K_f = data['K_f']         #radiator material heat transfer coefficient
        self.N_f = data['N_f']         #no of fins per meter
        self.N_p = data['N_p']         #no of fin profiles
        self.N_ct = data['N_ct']  #no of tube columns in one row
        self.N_r = data['N_r']         #no of tube rows
        self.B_h = data['B_h']         #core height
        self.B_w = data['B_w']         #core width
        self.B_t = data['B_t']         #core thickness
        self.F_p = data['F_p']         #fin pitch
        self.F_t = data['F_t']         #fin thickness
        self.F_h = data['F_h']         #fin height(distance between 2 pipes)#
        self.alpha_f = data['alpha_f'] #fin angle
        self.R_f = data['R_f']         #fin radius
        self.Y_cl = data['Y_cl']       #coolant tube length
        self.Y_cw = data['Y_cw']       #coolant tube width
        self.Y_t = data['Y_t']         #coolant tube thickness
        self.Y_l = data['Y_l']         #coolant tube length
        self.R_t = self.Y_cw/2         #coolant tube radius
        self.L_p = data['L_p']         #louver pitch
        self.L_l = data['L_l']         #louver length
        self.L_a = data['L_a']         #louver angle
        self.L_h = self.L_p * m.sin(self.L_a*m.pi/180)    #louver height
        self.T_d = self.Y_cl * self.N_r
        self.T_p = self.F_h + self.Y_cw         #coolant tube pitch
```

2) Developed a method to calculate other radiator parameters

```
def calculate_radiator_parameters(self):
    # F_l = total length of a fin
    # A_fr_r = radiator frontal area
    # A_fr_t = frontal area of tubes
    # A_fr_f = frontal area of fins
    # A_f = total fin area
    # A_a = air side heat transfer area
    # A_c = coolant side heat transfer area
    # A_pa = air flow minimum area

    self.F_l = m.pi*self.R_f + (self.F_h - 2*self.R_f)/m.cos(self.alpha_f*m.pi/180)
    self.A_fr_r = self.B_w * self.B_h
    self.A_fr_t = self.Y_l * self.Y_cw * self.N_ct
    self.A_fr_f = self.F_t * self.F_l * self.N_f * self.Y_l * self.N_p
    self.A_f = 2 * self.B_t * self.F_l * self.N_f * self.Y_l * self.N_p
    self.A_a = self.A_f + 2 * self.N_ct * self.Y_l * self.N_r * ((self.Y_cl - 2*self.R_t) +
2*m.pi*self.R_t)
    self.A_c = (2*m.pi*(self.R_t - self.Y_t) + 2*(self.Y_cl - 2*self.R_t))*self.Y_l*self.N_r*self.N_ct
    self.A_pa = self.A_fr_r - self.A_fr_f - self.A_fr_t
```

3) Developed a method to approximate overall heat transfer coefficient. we have tried to use different correlations to obtain air side heat transfer coefficient. After testing we decided to use [1]. Then the coolant side heat transfer coefficient was approximated by using Hansen's equation

```
def set_U(self, air, coolant):

# AIR SIDE CALCULATIONS

    rho_a = air['rho']
    c_p_a = air['c_pa']
    mu_a = air['mu']
    v_a = air['v']
    Pr_a = air['Pr']
    k_a = air['k']



    G = (self.A_fr_r*rho_a*v_a)/ self.A_pa
    Re_lp = G*self.L_p/mu_a

    # fan

    # Using Davenport's correlation 1983
    # J1 = 0.249 * Re_lp**(-0.42) * self.L_h**0.33 * (self.L_h/self.F_h)**1.1 * (self.F_h**0.26)

    # Using Dong, Chen correlation 2007
    J = 0.26712*Re_lp**(-0.1944) * (self.L_a/90)**0.257 * (self.F_p/self.L_p)**(-0.5177) *
(self.F_h/self.L_p)**(-1.9045) * (self.L_l/self.L_p)**(1.7159) * (self.B_t/self.L_p)**(-0.2147) *
(self.F_t/self.L_p)**(-0.05)
    f = 0.54486*Re_lp**(-0.3068) * (self.L_a/90)**0.444 * (self.F_p/self.L_p)**(-0.9925) *
(self.F_h/self.L_p)**0.5458 * (self.L_l/self.L_p)**(-0.2003) * (self.B_t/self.L_p)**(0.0688)
    self.J = J
    self.Re_lp = Re_lp

    # using Chang's 1997 correlation
    # J3 = Re_lp**(-0.49) * (self.L_a/90)**0.27 *  (self.F_p/self.L_p)**(-0.14) * (self.F_l/self.L_p)**(-
0.29) * (self.T_d/self.L_p)**(-0.23) * (self.L_l/self.L_p)**(0.68) * (self.T_p/self.L_p)**(-0.28) *
(self.F_t/self.L_p)**(-0.05)

    h_a = J * G * c_p_a / Pr_a**(2/3)
    self.h_air = h_a

    l = self.F_h/2
    m_eff = m.sqrt(2*h_a/(self.K_f*self.F_t))
    # print('meff', m_eff)
    eff_f = m.tanh(m_eff*l)/(m_eff*l)
    eff_o_a = 1 - (1-eff_f)*self.A_f/self.A_a


# coolant side calculations

    rho_c = coolant['rho']
    mu_c = coolant['mu']
    Pr_c = coolant['Pr']
    m_dot_c = coolant['m_dot']
    k_c = coolant['k']
    m_dot_c_t = m_dot_c/(self.N_ct*self.N_r)


    A_ff_t = (self.Y_cl - self.Y_cw)*(self.Y_cw - 2*self.Y_t) + m.pi*((self.Y_cl - 2*self.Y_t)/2)**2 #
freeflow area per tube
    P_ff_t = 2*(self.Y_cl - self.Y_cw) + m.pi*(self.Y_cw - self.Y_t*2)
    D_h_t = 4*A_ff_t/P_ff_t

    G_c = m_dot_c_t/A_ff_t
    Re_c = G_c * D_h_t/mu_c

    # Hansen equations --> mean nusselt number
    Nu_c = 3.66 + (0.0668 * (D_h_t/self.Y_l) * Re_c * Pr_c)/(1 + 0.04*((D_h_t/self.Y_l)*Re_c*Pr_c)**(2/3))
    # Nu_c = 0.023*Re_c**0.8*Pr_c**0.4


    h_c = k_c * Nu_c/D_h_t

# overall heat tranfer coefficient

    R = 1/(eff_o_a*h_a*self.A_a) + (1/(h_c*self.A_c))
    self.UA = 1/R
```

4) Then the overall heat transfer of the radiator was calculated using NTU – effectiveness method.

```
def NTU(self, air, coolant):
    rho_a = air['rho']
    c_p_a = air['c_pa']
    mu_a = air['mu']
    v_a = air['v']
    Pr_a = air['Pr']
    k_a = air['k']
    T_i_a = air['T_i_a']
    # T_o_a = air['T_o_a']

    rho_c = coolant['rho']
    mu_c = coolant['mu']
    Pr_c = coolant['Pr']
    m_dot_c = coolant['m_dot']
    k_c = coolant['k']
    c_p_c = coolant['c_pc']
    T_i_c = coolant['T_i_c']


    m_dot_a = self.A_fr_r * v_a * rho_a
    C_a = m_dot_a * c_p_a
    C_c = m_dot_c * c_p_c

    C_min = min(C_a, C_c)
    C_max = max(C_a, C_c)
    q_max = C_min * (T_i_c - T_i_a)

    NTU = self.UA/C_min
    Cr = C_min/C_max

    eff = 1 - m.exp((NTU**0.22)*(m.exp(-Cr * NTU**0.78) - 1)/Cr)
    self.q = q_max*eff
```

5) The air and water properties are defined like this to be used in the simulation.

```
# at 30
air = {
    'rho':1.16,#1.16, # dry air at 1 atm
    'mu': 1.9e-5, #pas
    'c_pa': 1007,  #J/m2K
    'v': 50, # m/s
    'Pr': 0.7,
    'k': 0.0270,
    'T_i_a': 33,
}

# at 50C
water = {
    'rho': 1000, #kg/m-3
    'mu': 0.0005474, #Pa.s
    'Pr': 5,
    'm_dot': 0.2, #kg/s
    'c_pc': 4180, #j/kgK
    'k': 0.64060,  #W/mK
    'T_o_c': 45,
    'T_i_c': 55,
}
```

Code validation

To validate the air side heat transfer coefficient and J factors from the code we have used the data set provided in [1]. We used one set of fin characteristics provided in the paper and calculated h and J from our code and compared it against the results provided in the paper.

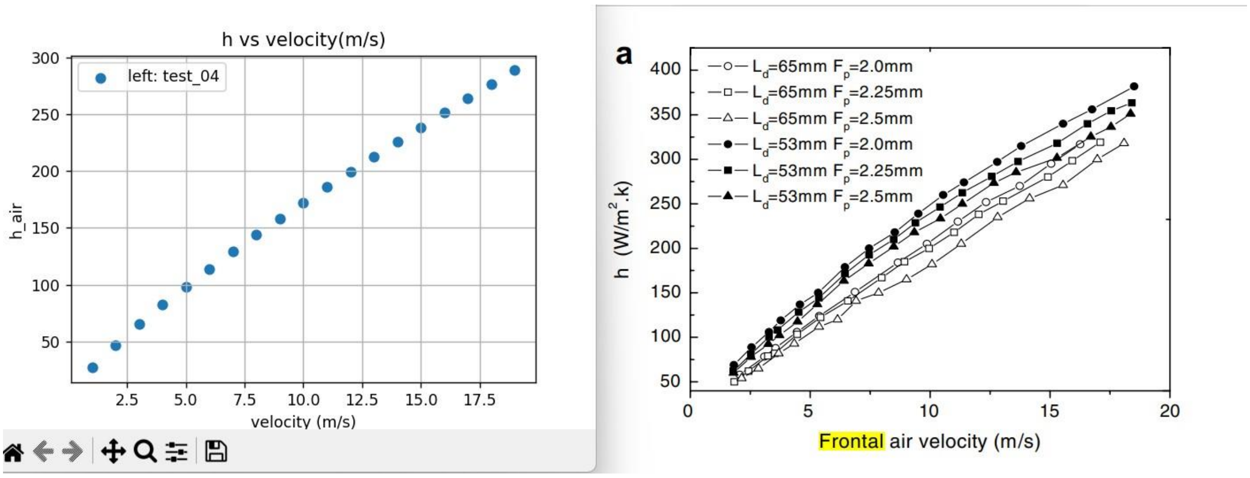

Figure 02: On the right is the results for heat transfer coefficient for air obtained from the code and on right published on paper [1]. We have used the 4th data set in our code

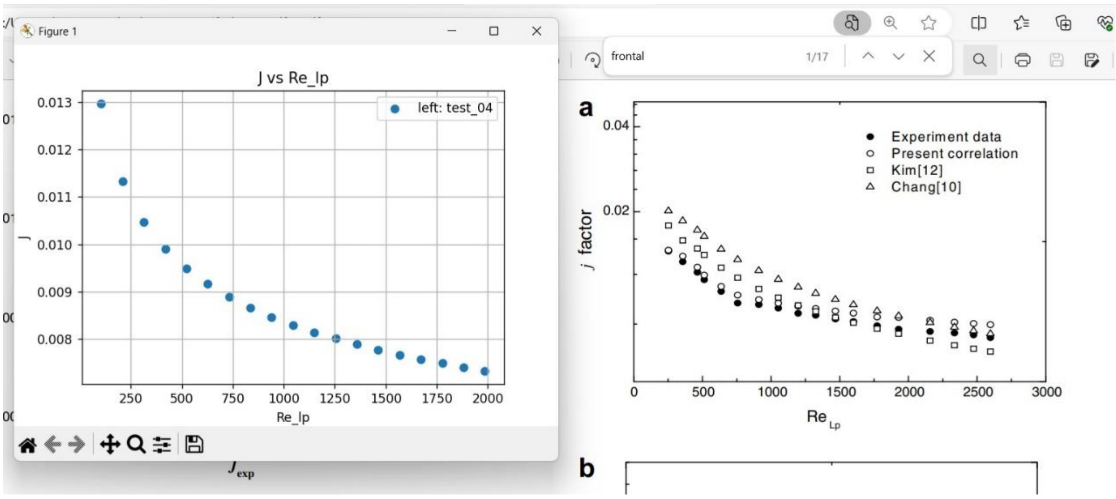

Figure 03: on right the J factor data from the code and on left result published on the paper [1]

Validating heat dissipation rate using LMTD method

Let's consider the arrangement in figure 06. The DRZ400 radiator data was selected for this test. Ambient air temperature was set to 33 C and air frontal velocity was set to 20 m/s. The calculated coolant outlet temperature of the radiator was measured for 100 iterations. The system can be considered as stable after 100 iterations according to figure 04.

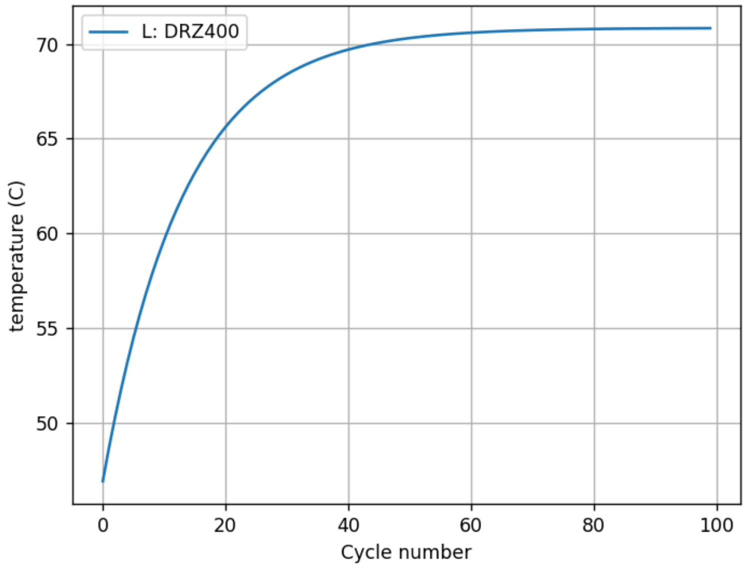

Figure 04: radiator coolant outlet temperature variation with each iteration

At the 100th iteration according to the simulation,

$$coolant\ radiator\ outlet\ temperature(T_{h2}) = 70.8\ °C$$
$$coolent\ radiator\ inlet\ temperature(T_{h1}) = 73.0\ °C$$
$$air\ mass\ flow\ rate\ (m) = 0.62\frac{kg}{m^3}$$
$$heat\ dissipation\ (q) = 2500\ W$$
$$air\ outlet\ temperature\ (T_{c2}) = don't\ know$$
$$ambient\ temperature\ (T_{c1}) = 33\ °C$$

Considering heat energy received by the air,

$$q = mc_{p,a}\Delta T$$
$$2500 = 0.62 \times 1007 \times (T_{c2} - 33)$$
$$T_{c2} = 37\ °C$$

Using LMTD method,

$$q = UAF\Delta T_{LMTD} \qquad \text{------- (4)}$$

$$\Delta T_{LMTD} = \frac{(T_{h2} - T_{c2}) - (T_{h1} - T_{c1})}{\ln\left(\frac{T_{h2} - T_{c2}}{T_{h1} - T_{c1}}\right)}$$
$$= 37\ °C$$

To find the correction factor F,

$$P = \frac{T_{c2} - T_{c1}}{T_{h2} - T_{c1}}$$
$$P = 0.098 \approx 0.1$$

$$R = \frac{T_{h2} - T_{h1}}{T_{c2} - T_{c1}}$$
$$= 0.925$$

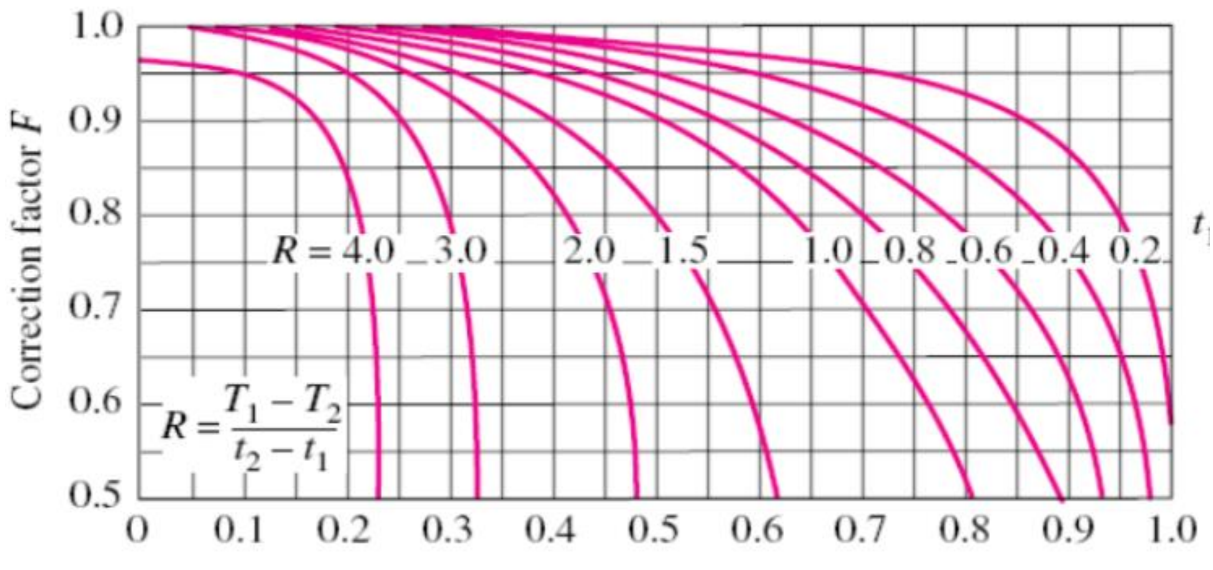

Figure 05: F factor variation with P and R for crossflow one pass 2 fluids unmixed heat exchanger

According to figure 05, F can be considered to be 1,

Then from (4),

$$2500 = UA \times 1 \times 37$$
$$UA = 67.56\ W/K$$

The UA value we obtained from the code was 68.82 W/K. hence this code can be validated.

SELECTION OF A RADIATOR

Series of analytical tests was carried out to find a suitable radiator option for our design. The main goal here is to identify the better radiator. We have considered the maximum power loss for these calculations

Conditions:

- Coolant mass flowrate = 0.2 kg/s
- Radiator outlet coolant temperature = 45 C

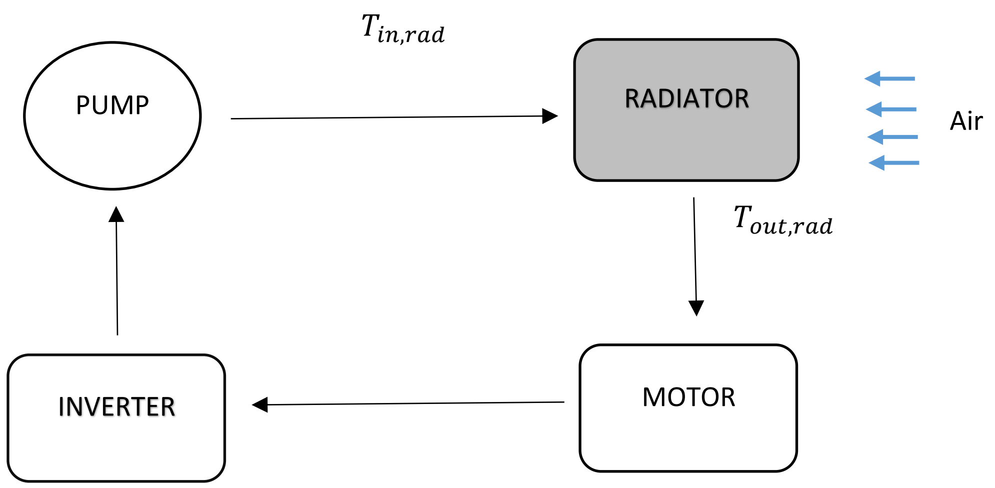

Figure 06: Schematic of the test

The radiator coolant inlet temperature was calculated using simple $q = mc\Delta T$ equations.

$$T_{in,rad} = 54.42\ °C$$

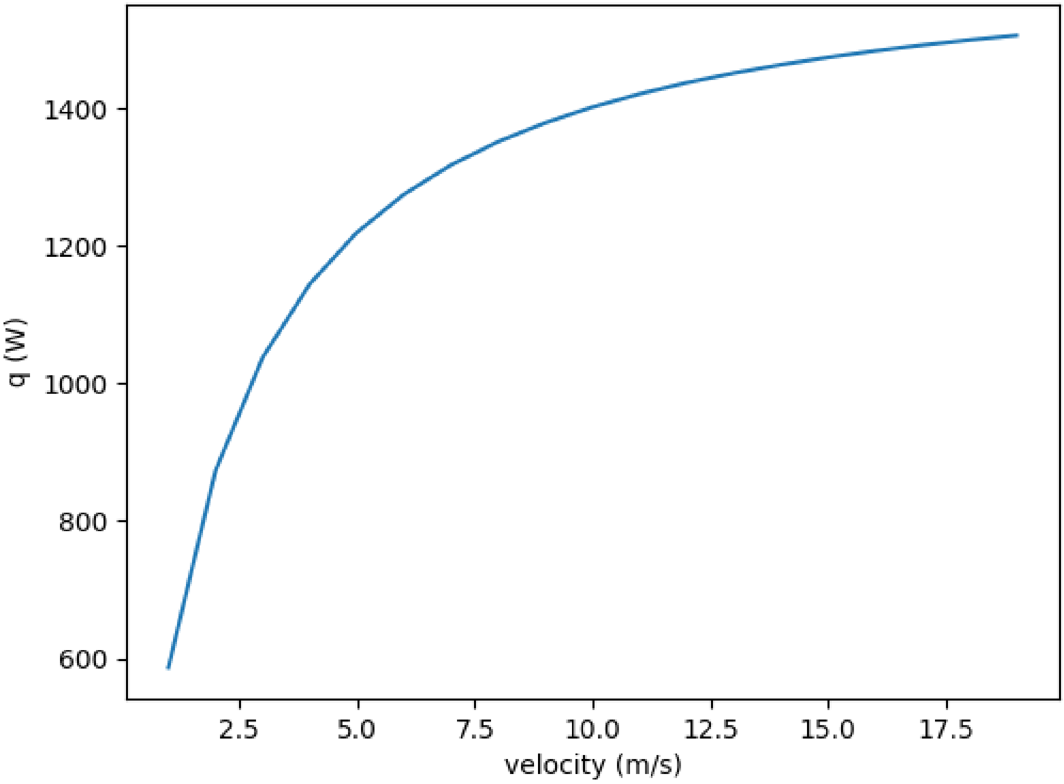

Figure 07: heat dissipation rate variation with the velocity for DRZ400 single radiator

It should be noted that the heat transfer rates we are obtained are not exactly very accurate values since the approximation and assumptions we have done. But they will be enough for us to select the better arrangement by see the variations with different parameters.

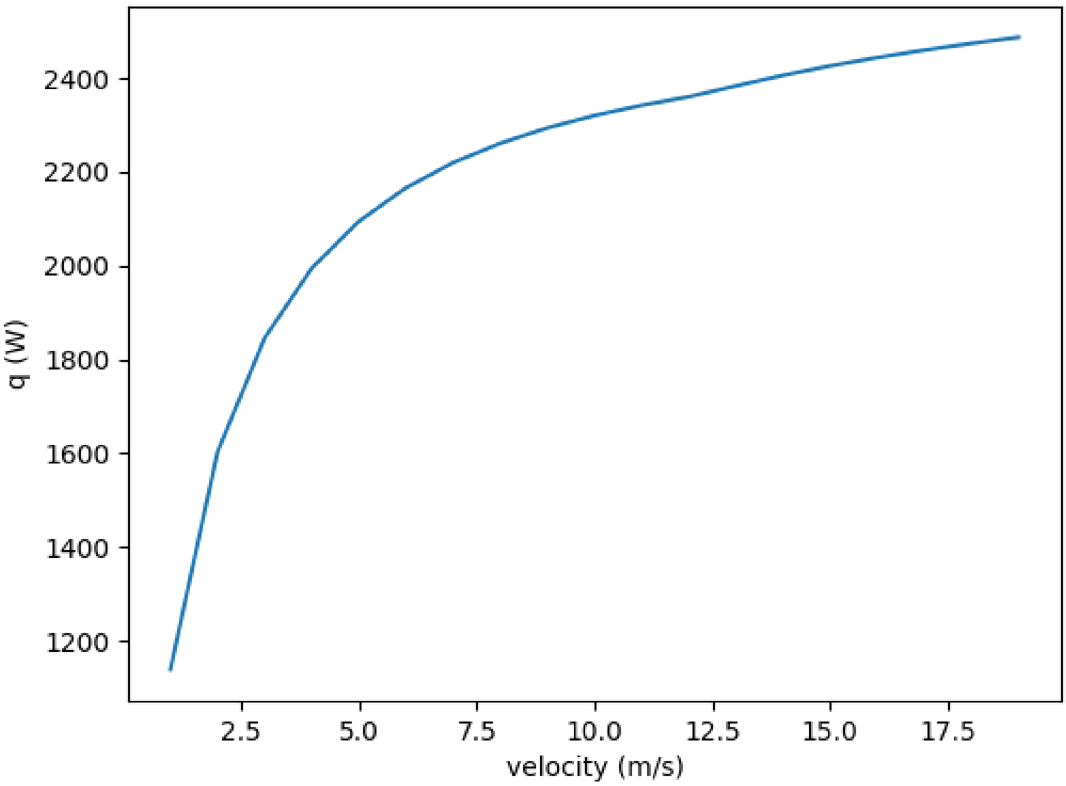

Figure 08: heat dissipation rate variation with the velocity for YFM700r single radiator

The heat transfer capacity of the YMF700R is about 1.6 times as the DRZ400 radiator. But it is almost two times bigger than the DRZ400 radiator. Both has advantages and disadvantages. We've planned to use DRZ400 left and right radiator combination instead side pods. This arrangement will provide a better weight distribution. Also it is aerodynamically advantageous since ducting can be done on the both sides to improve the air flow rate.

Hence we have decided to use DRZ400 left & right radiator combinations in the side pods.

SYSTEM CONFIGURATION

two DRZ400 Radiators are connected in parallel.

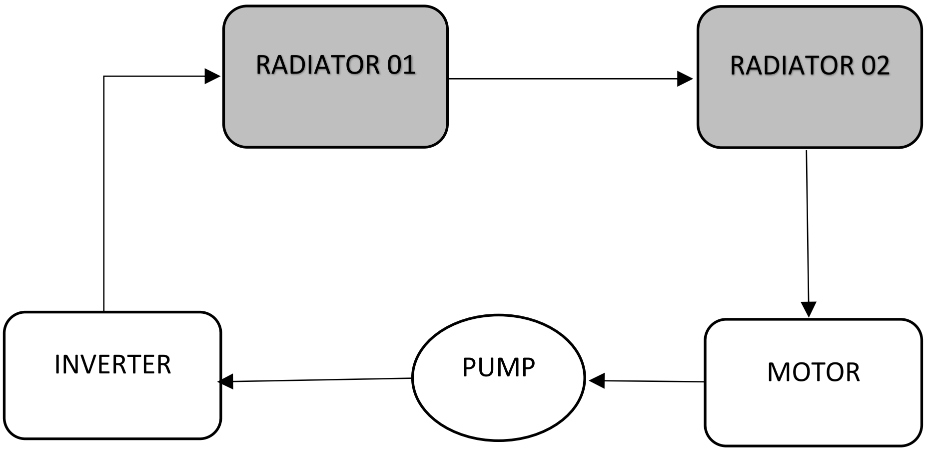

Figure 09: Schematic of the second configuration

To simulate this configuration we have considered

- Air properties at 33℃
- Water properties at 50℃
- Average power loss

We have considered the average power loss since the maximum power loss is not continuous. Designing system for the maximum rate will end up making the system larger and heavier. Since this is a performance car, we decided to design the system to accommodate average heat dissipation of 2.8 kW.

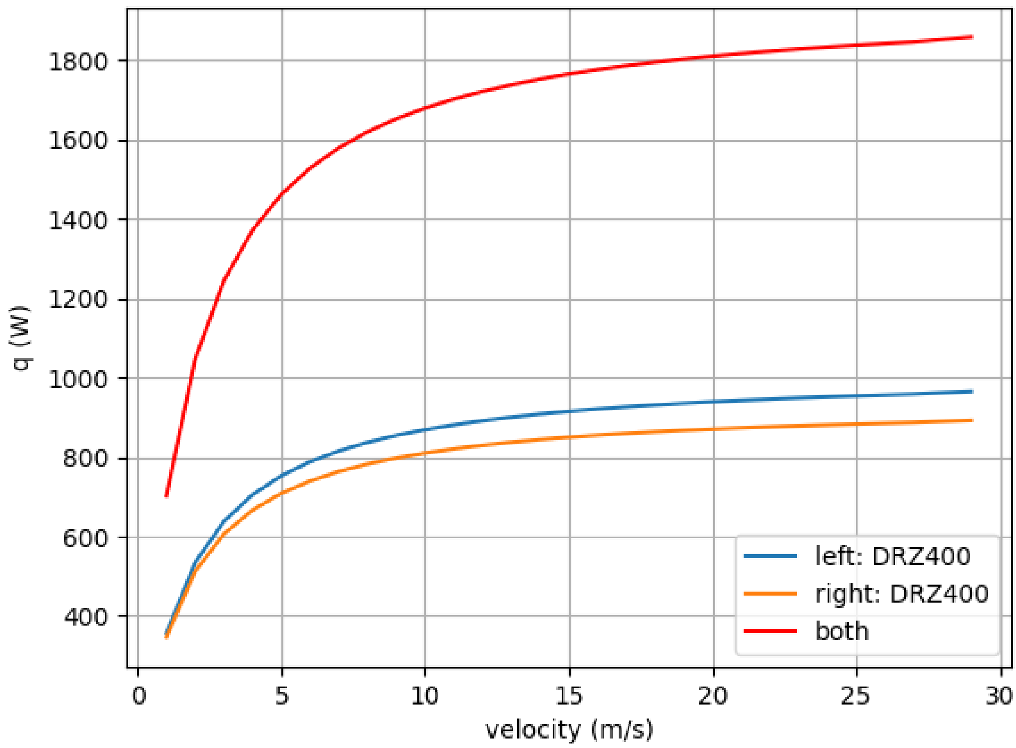

Figure 10: heat dissipation rate vs velocity

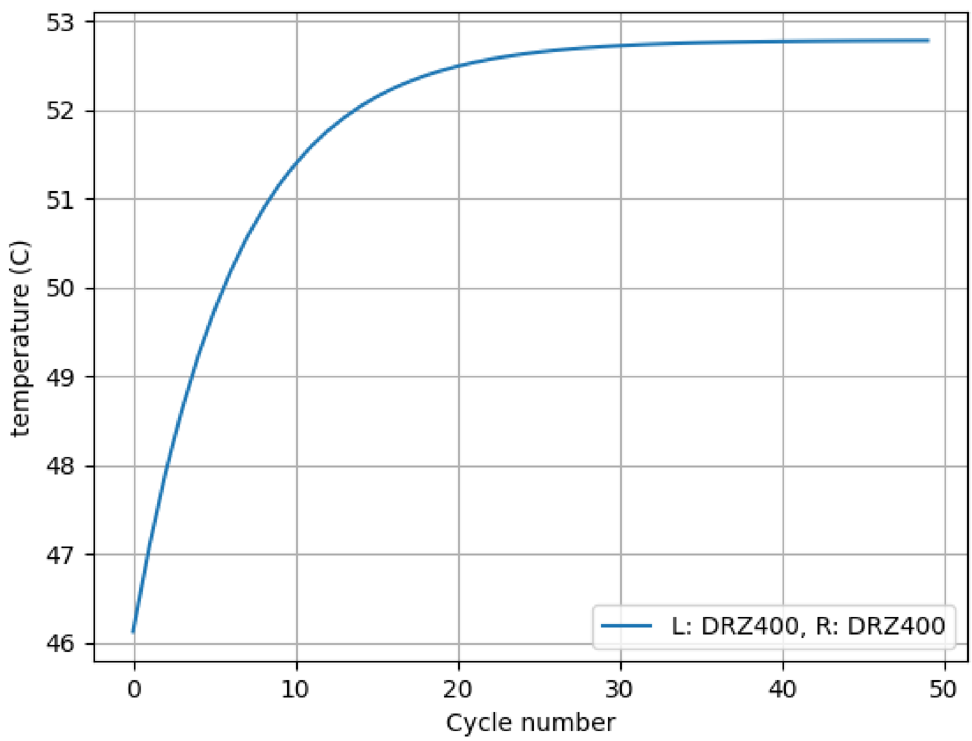

Figure 11: Coolant temperature variation at radiator outlet with cycles

According to the figure 10 we cannot dissipate total 2.7 kW when the coolant operates at the 45℃ with an ambient temperature of 33℃ to analyze the worst case. Because of it the coolant temperature raises up to 53 C according to figure 11. In ideal conditions this will maintain the coolant radiator output of the system under 50 C like below.

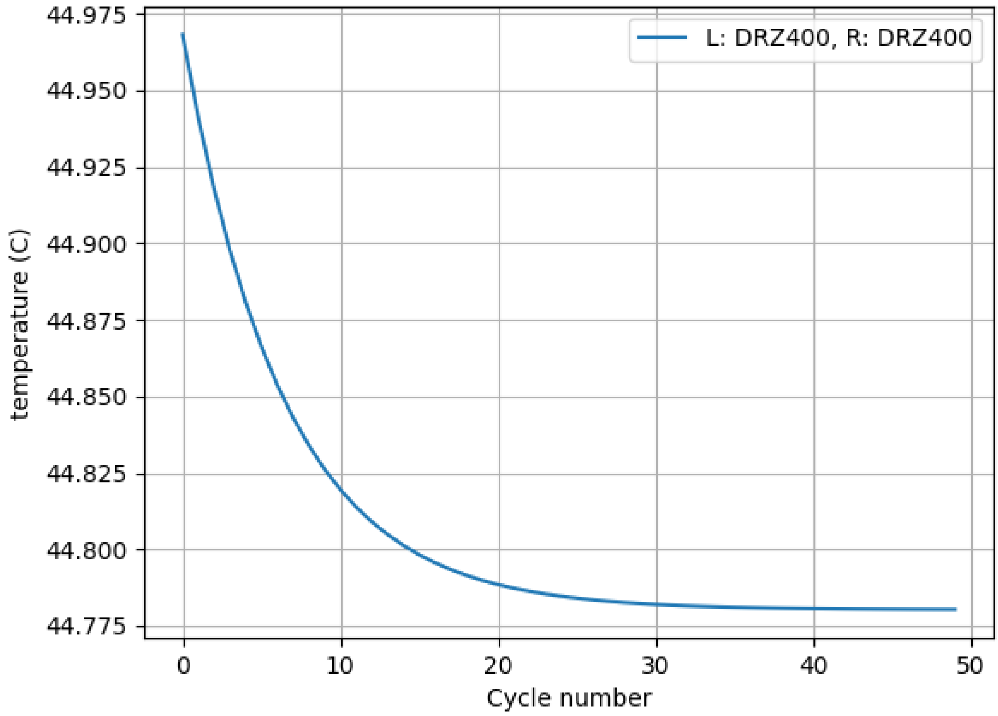

Figure 12: Coolant temperature at radiator outlet variation at ambient temperature of 25 C

PUMP SELECTION

Major pressure drops in the system

- Motor = 0.3 bar (data sheet)
- Inverter = 0.6 bar (data sheet)
- Radiator = 0.04 x 2 = 0.08 bar (calculated)
- Pump loss = 10%

$$P_{loss} = 0.98 + 0.1 = 1.08\ bar$$

With the increase in flow rates the above calculated pressure can be changed. So 1.5 safety factor was introduced.

$$P_{total\ loss} = 1.6\ bar$$

Pump requirement:

Should be able to pump water at rate of 12 l/min with 1.5 bar pressure.

For the pump Everflo 12V EF4000 diaphragm pump was selected since it can pump 12LMP with 2.8 bar pressure

EF4000:

BAR	PSI	MPa	LPM	GPM	Amps
0.5	7	0.05	15.1	4.0	6.0
1.0	15	0.10	14.3	3.8	7.6
1.5	22	0.15	13.6	3.6	8.9
2.0	29	0.20	12.8	3.4	10.0
2.5	36	0.25	12.0	3.2	11.0
3.0	44	0.30	11.3	3.0	12.0
3.4	49	0.34	10.5	2.8	12.7
4.1	59	0.41	-	-	-

Figure 13: Everflo EF4000 pump chart

PIPING

Near the radiator we have to ensure the structural rigidity and increase the heat transfer rate by using Aluminum tubes.

Inside the vehicle we have used silicon tubes to ensure insulation. And also these pipes are cost effective.

According to the low flow rate the velocities inside pipe flow will also have a low Reynolds number. Hence the pipe flow will be laminar. According to Darcy's equation pressure drop through the pipe will be at lowest when the velocity is also low. Hence bigger the pipe diameter is better. According to the data sheets EMRAX228 has 10 mm fitting while BAMOCAR D3 has 13 mm.

Table 04: Selected components

Right radiator	DRZ400
Left radiator	DRZ400
Pump	EverFlo EF4000
Pipes	Aluminum and Silicon

CATCH CAN

A catch can will be implemented according to the Formula Bharat rule book. At this moment we are unable to specify the capacity. It will be design and manufactured in house.

REFERENCES

1. Dong, J., Chen, J., Chen, Z., Zhang, W.-F. and Zhou, Y. (2007). Heat transfer and pressure drop correlations for the multi-louvered fin compact heat exchangers. Energy Conversion and Management, 48(5), pp.1506–1515

2. Chang YJ, Wang CC. A generalized heat transfer correlation for louver fin geometry. Int J Heat Mass Transfer 1997;40(3):533–44

3. Compact heat exchangers, W. M. Kays and A. L. London, Third edition

4. P W Johnson, S Watkins. (2005). An analytical study on heat transfer performance of radiators with non-uniform airflow distribution