

Load Characteristic MOTORC

Load Characteristic MOTORC *Dynamic Phasor Model of a single-phase induction motor load model* *With Block Under-Voltage Tripping, Contactor, and Thermal Relay*

Saturation Equation:

$$Sat = \begin{cases} 1 & \text{for } |\psi| \leq B_{sat} \\ 1 + A_{sat}(|\psi| - B_{sat})^2 & \text{for } |\psi| > B_{sat} \end{cases}$$

This model includes a thermal relay, 2-stage definite time under-voltage relay, and a contactor relay which are the same as those used with the LD1PAC model. The implementation of all these relaying features will be described below. The output of the phasor model for the single-phase motor will be a current into the network (I_s) which is described in the INDMOT1P model. All the relaying features will be implemented as a multiplication of this current by a factor which is a function of these relaying outputs. This model is meant to model 1000s of individual single-phase motors with a single set of dynamic equations, so the only thing we can do is add this simplistic multiplier.

The total fraction of stator current modeled at the network interface multiplied by the factors associated with the 3 different relaying possibilities. The calculations of the factors f_{uvr} , K_{con} , and K_{th} are described in future sections too.

Using the contactor and definite time undervoltage relaying type system used in LD1PAC, we calculate the following

$$\begin{aligned} K_{uv} &: \text{fraction of motors that have not been tripped by undervoltage relays} \\ K_{con} &: \text{fraction of motors that have not been tripped by contactor relays} \\ K_{th} &: \text{fraction of motors that have not been tripped by thermal relay} \\ K_{uv}K_{con}K_{th} &: \text{Total fraction of motors that have not been tripped by any of the relays} \end{aligned}$$

This means that the current modeled at the network interface in the network boundary equations will be equal to $I_{s\ network} = I_s K_{uv} K_{con} K_{th}$ where the value I_s will be described in the equations for the rotor and stator later in this document.

Within the code, the special feature is that the function "FractionOfNominalStillOn" will now be used. This is only used in the network boundary equations portion of the simulation. We continue to simulate a single set of equations for a single phase induction motor, but the relays are applied on top of this as a fractional multiplier in the network boundary equations where the fractional multiplier is between 0.0 and 1.0.

```
function FractionOfNominalStillOn : double;  
begin  
    result := TwoBlockUnderVoltageTripping.fKuv  
        * ContactorTripping.fKcon  
        * ThermalRelay.Kth  
end;
```

This model is meant to model 1000s of individual single-phase motors with a single set of dynamic equations, so the only thing we can do is add this simplistic multiplier.

1 Thermal Overload Protection Heating (Kth)

T_{th} Compressor heating time constant in sec.

$Th1t$ Compressor motors begin tripping

$Th2t$ Compressor motors finished tripping

This model describes the stator currents in a dq reference frame, so to get the I^2R heating for the motor we need to add the heating in both the d and q axis.

$$Heat = I^2R = (I_{dsR}^2 + I_{dsI}^2)R_{ds} + (I_{qsR}^2 + I_{qsI}^2)R_{qs}$$

We are going to be simulating everything in a forward/back reference frame instead though, so we'll express the heating in terms of forward and backward current instead using:

$$I_{dsR} = (I_{fR} + I_{bR})$$

$$I_{dsI} = (I_{fI} + I_{bI})$$

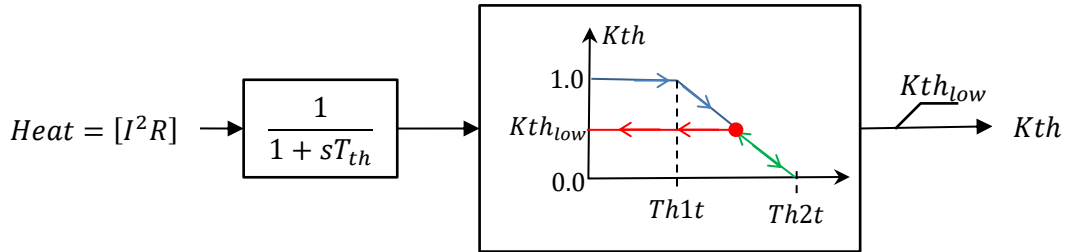
$$I_{qsR} = \frac{1}{n}(-I_{fI} + I_{bI})$$

$$I_{qsI} = \frac{1}{n}(I_{fR} - I_{bR})$$

Substitute these into our dq currents and we get the following function for heating.

$$Heat = [I^2R] = [(I_{fR} + I_{bR})^2 + (I_{fI} + I_{bI})^2]R_{ds} + \left[\frac{1}{n^2}(-I_{fI} + I_{bI})^2 + \frac{1}{n^2}(I_{fR} - I_{bR})^2\right]R_{qs}$$

This heating value is calculated in the MVABase of the load in per unit. The heating value is fed into a simple delay block with thermal time constant of T_{th} . If the temperature is below $Th1t$, then the remains in service load, but starting at $Th1t$ a fraction of the load will be tripped and the fraction tripped will linearly increase up until at a temperature of $Th2t$ at which all the motor is tripped. Once a portion of the load has tripped due to the thermal relay it will not restart during the simulation (software keeps track of Kth_{low} during the simulation.)



Each time step code is written to calculate a new Kth output of the thermal relay model. The value below "theTemp" is the system state representing the present temperature (output of the delay block).

2 Under Voltage Relays (1-fuvr)

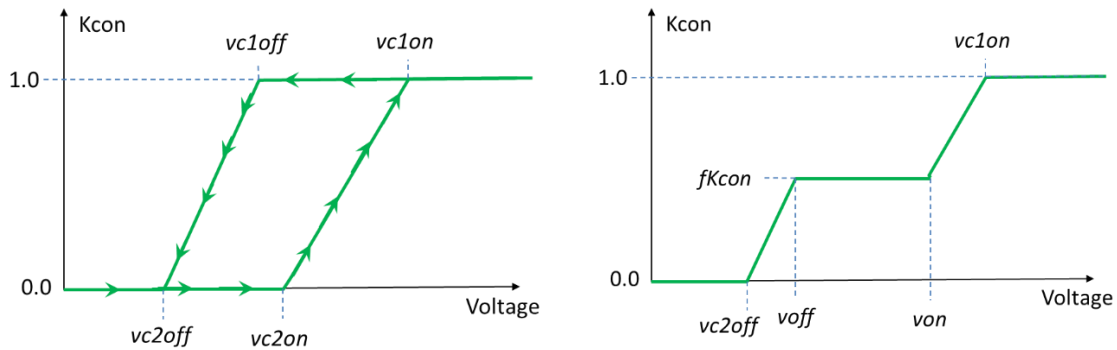
T_v	Voltage measurement input delay in seconds (This filtered voltage is used as an input to the contactor model and the undervoltage relays only)
V_{c1off}	Voltage 1 contactor disconnect load in p.u.
V_{c2off}	Voltage 2 contactor disconnect load in p.u.
V_{c1on}	Voltage 1 contactor re-connect load in p.u.
V_{c2on}	Voltage 2 contactor re-connect load in p.u.

If the measured voltage (after T_v measurement time constant) drops below $uvtr1$ per unit for T_{tr1} seconds or below $uvtr2$ per unit for T_{tr2} seconds, then a fraction of the ($fuvr$) will be tripped. Once this fraction is tripped it will not be reconnected. When the relay trips we set the parameter $Kuv = 1 - Fuvr$.

3 Contactor Relay (Kcon)

T_v	Voltage measurement input delay in seconds (This filtered voltage is used as an input to the contactor model and the undervoltage relays only)
f_{uvr}	Fraction of compressor motors with undervoltage relays
$uvtr1$	Undervoltage pickup level 1 in p.u.
T_{tr1}	Definite time for undervoltage pickup level 1 in seconds
$uvtr2$	Undervoltage pickup level 2 in p.u.
T_{tr2}	Definite time for undervoltage pickup level 2 in seconds

The contactor model represents a model of how air conditioners trip off as the voltage falls and then turn back on as the voltage increases. They start tripping off at a voltage of $vc1off$ and then fall off linearly until all are completely off at $vc2off$. They then start turning back on at V_{c2on} and increase linearly until all are back on at $vc1on$. This is shown on the left curve below. To be more explicit regarding the way this is handled in the code though, at the beginning of each time step a value of $Kcon$ is calculated and values for $voff$ and von at which the respective curves are intercepted. Once $Kcon$ drops below 1.0 then as the voltage moves back up and down it will move at a constant $Kcon$ level until it hits the other line as shown in the figure on the right.



The voltage used in the contactor function (x-axis of curves to the right) is the measured voltage after the T_v measurement time constant.