

Load Characteristic INDMOT1P_PTR

Load Characteristic INDMOT1P_PTR Dynamic Phasor Model of a single-phase induction motor load model With Progressive Tripping and Reconnecting

This model builds on the INDMOT1P model, but makes some modifications

1. Removes the modeling of magnetic saturation. The aSat term in the INDMOT1P model is assumed to be 1.0 for the entire simulation (no saturation)
2. Simplifies the mechanical torque equation to a single parameter Etrq
3. Models progressing tripping and reconnecting for under voltage and a thermal relay

Parameter	Parameter Description
LF	Load Factor. All values are on an $MVABaseUsed = \frac{MW_{init}}{LF}$
$MBase$	If $MBase > 0$ then machine parameters are on this $MVABaseUsed$ is equation to this, otherwise the $MVABaseUsed$ is MW_{init}/LF
T_{nom}	Nominal load torque used for motor starting studies. For online motors this is automatically calculated from initial conditions.
R_{ds}	stator d-axis resistance in per unit
X'_{ds}	stator d-axis reactance in per unit
R_{qs}	stator q-axis resistance in per unit
X'_{qs}	stator q-axis reactance in per unit
X_c	capacitor reactance in per unit (will be negative)
X_m	magnetizing reactance in per unit
n	ratio of stator auxiliary winding turns to stator main winding turns
H	inertia constant for the motor
R_r	rotor resistance in per unit that is to calculate $T'_o = \frac{X_r}{\omega_b R_r}$
X_r	rotor reactance in per unit
E_{trq}	Exponential term of the mechanical torque equation: $T_{mech} = T_{nom} \left(\omega_r^{E_{trq}} \right)$
T_{th}	Compressor heating time constant in seconds
T_{h1t}	Compressor motors begin tripping
T_{h2t}	Compressor motors finish tripping
T_v	Voltage measurement delay in seconds
V_{1off}	Voltage in per unit at which load fraction begins decreasing
V_{2off}	Voltage in per unit at which load fraction decreases to zero
V_{1on}	Voltage in per unit at which load fraction begins increasing
V_{2on}	Voltage in per unit at which load fraction increases back to $[FracMin + Frecon * (1.0 - FracMin)]$
F_{recon}	Fraction of load that has been disconnected that will come back as voltage recovers
T_{delay}	Time delay use to approximate time vs voltage nature of load loss as the voltage is decreasing expressed in seconds
V_{td}	Voltage threshold below with the Time delay used starts decreasing toward 0 seconds at a voltage of 0
T_{recon}	Time delay use to approximate the reconnection of load as the voltage is increasing

The only difference with this model is that a Progressive Tripping and Reconnection equations are modeled as well as a thermal temperature relay. The structurally 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 progressive tripping and reconnecting block and the thermal relay 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 := ProgressiveVoltageRelay.FracOutput * 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.

The total fraction of stator current modeled at the network interface multiplied by the factors associated with the 2 different relaying possibilities. The calculations of the factors $Frac_{progressive}$ and K_{th} are described on the next pages.

$Frac_{progressive}$: fraction of presently active motors based on progressive tripping and reconnecting block
 K_{th} : fraction of motors that have not been tripped by thermal relay
 $Frac_{progressive}K_{th}$: Total fraction of motors that have not been tripped by any of the relays

This means that the current modeled at the network interface in the network boundary equations will be equal to $I_{s\ network} = I_s Frac_{progressive} K_{th}$ where the value I_s is what the current injection would be the simulated load.

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:

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

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

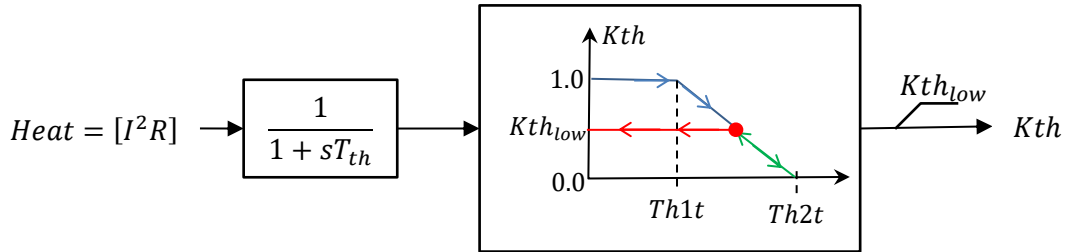
$$I_{qsR} = \frac{1}{n}(-I_{fI} + I_{bI})$$

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

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

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

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.)

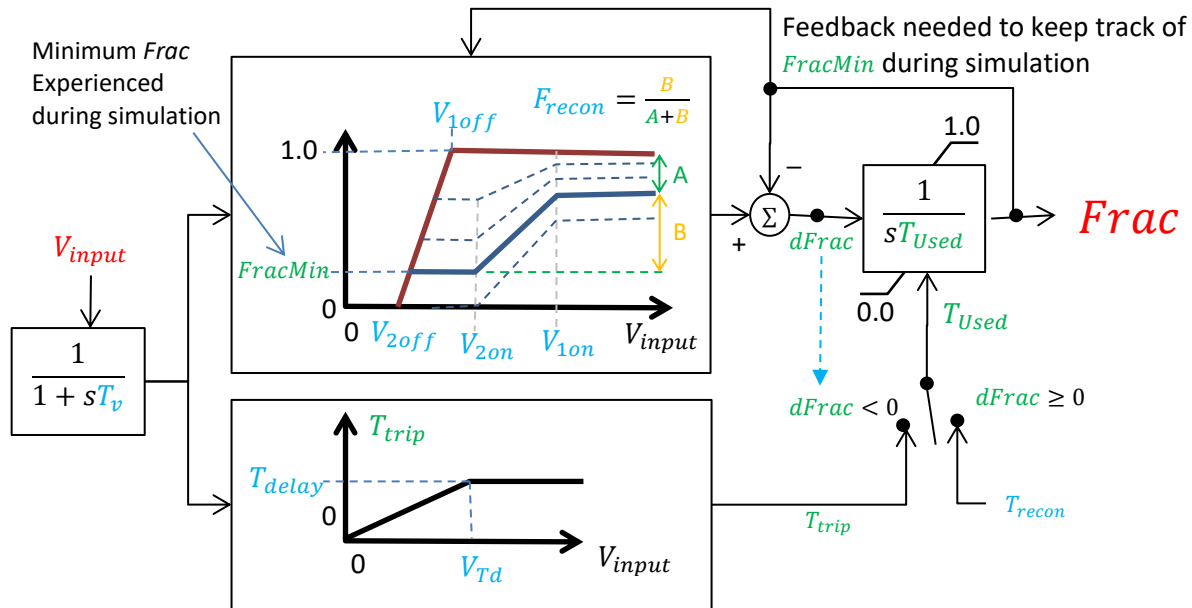


Each time step code is written to calculate a new Kth output of the thermal relay model. The value below "theTemp" is the system state representing the present temperature (output of the delay block).

2 Progressive Tripping and Reconnecting Block

T_v	Voltage measurement delay in seconds
V_{1off}	Voltage in per unit at which load fraction begins decreasing
V_{2off}	Voltage in per unit at which load fraction decreases to zero
V_{1on}	Voltage in per unit at which load fraction begins increasing
V_{2on}	Voltage in per unit at which load fraction increases back to $[FracMin + Frecon \cdot (1.0 - FracMin)]$
$Frecon$	Fraction of load that has been disconnected that will reconnect as voltage recovers
T_{delay}	Time delay use to approximate time vs voltage nature of load loss as the voltage is decreasing expressed in seconds
V_{td}	Voltage threshold below with the Time delay used starts decreasing toward 0 seconds at a voltage of 0
T_{recon}	Time delay use to appproximate the reconnection of load as the voltage is increasing

A generic block for use in other models within PowerWorld Simulator will be created. This block may have the following 8 input parameters: T_v , V_{1on} , V_{1off} , V_{2on} , V_{2off} , $Frecon$, T_{delay} , V_{Td} and T_{recon} which are used as described in the following block diagram.



The input to this block diagram will be a voltage at the terminal bus. This will then be passes through a measurement delay block with time constant T_v . The output of this block will represent the presently active fraction of the load ($Frac$).

Tripping Characteristic: The parameters V_{1off} and V_{2off} describe the tripping characteristic. When a voltage input signal drops below V_{1off} and towards V_{2off} the output value of $Frac$ load begins dropping depending on the time constant on the output delay block that uses T_{Used} discussed shortly.

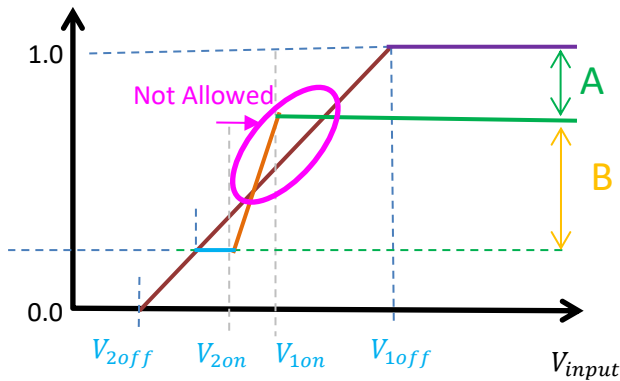
Reconnecting Characteristic: During the simulation, the block will also keep track of the minimum value of $Frac$ seen during the simulation. This calculated value $FracMin$, along with the parameters V_{1on} , V_{2on} , and $Frecon$ determine the shape of the dark blue curve above which represents the restarting characteristic. The parameter $Frecon$ represents what fraction of the load that has already tripped that

will restart after tripping. When the input is above V_{2on} the fraction will begin increasing until it reaches its maximum value of $(FracMin + (1 - FracMin) * F_{recon})$ at a voltage of V_{1on} .

Approximation of Time Delay: The parameters T_{delay} and V_{Td} control how quickly the fraction will decrease (tripping) when the voltage is dropping ($dFrac < 0$). The parameter T_{recon} controls how quickly the fraction will increase (reconnecting) when the voltage is recovering ($dFrac \geq 0$).

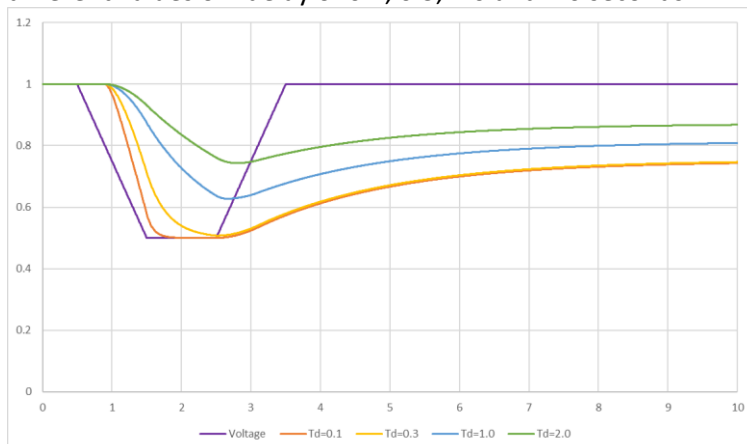
Restrictions are placed on the input parameters as follows

- $V_{2off} \leq V_{1off}$ (requirement so that the tripping characteristic makes sense)
- $V_{2off} \leq V_{2on} \leq V_{1on}$ (requirement so that the restarting characteristic makes sense)
- $V_{1on} \geq V_{1off}$ (this is to prevent the tripping and restarting characteristic from intersecting which would not make sense. See the following picture for an explanation)



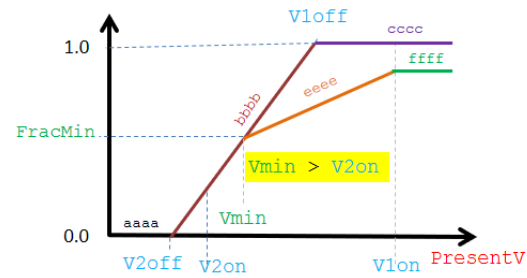
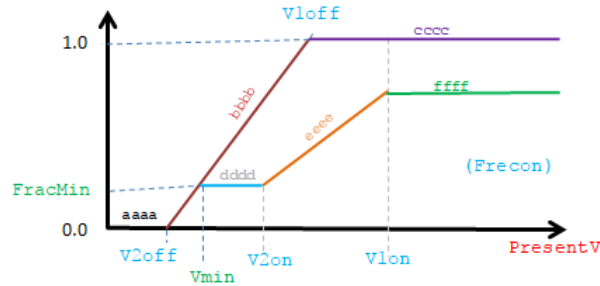
This block can be simplified by setting $V_{2off} = V_{2on}$ and $V_{1off} = V_{1on}$ which gives you a tripping and reconnecting characteristic which is similar to what has been done with the existing PVD1 and LDELEC models. The big difference with this new progressing tripping and reconnecting block is the introduction of the time delays and the tracking of the $FracMin$. The PVD1 and LDELEC models did not include any of this time response.

We tested out these ideas within a simple test signal. The following figure shows a purple line that represents the input voltage to our Progressive tripping and reconnecting block. The signal goes from 1.0 at 0.5 second linearly down to 0.5 per unit at 1.5 seconds. Then at 2.5 seconds it starts going back up linearly to 1.0 at 3.5 second. The parameters for this block were chosen as $V1off=V1on=0.8$; $V2off=V2on=0.2$; $F_{recon} = 0.5$; $V_{td}=0$; $T_{recon} = 2.0$. Finally, the four different color plot series represent different values of T_{delay} of 0.1, 0.3, 1.0 and 2.0 seconds.



2.1.1 Pseudo code for Progressive Tripping Reconnecting for Low Voltage Only

The pseudo-code for implementing the low voltage progressive tripping and reconnecting block follows. The following images are helpful for understanding how the pseudo code and line segments it is tracing.



User Parameters

Vloff, V2off, Vlon, V2on, Frecon

Rules:

```
V2off <= Vloff
Vlon >= Vloff
V2off <= V2on <= Vlon
```

Inputs to Block

DelayBlockOutput = output of Time Delay block
PresentV = measured voltage to block

Initialization section does the following

```
if Frecon < 0.0 then Frecon = 0.0
else if Frecon > 1.0 then Frecon = 1.0
// Order of precedence for trustworthiness of input is Vloff, V2off, V2on, then Vlon.
if V2off > Vloff then V2off = Vloff // decrease V2off to at least Vloff
if Vlon < Vloff then Vlon = Vloff // increase Vlon to at least Vloff
if V2on < V2off then V2on = V2off // increase V2on to at least V2off
if Vlon < V2on then Vlon = V2on // increase Vlon to at least V2on
Vmin = Vloff
FracMin = 1.0
```

Following updated at the beginning of each Time Step

```
// FracMin is the minimum fraction during the simulation
if FracMin > DelayBlockOutput < then begin
FracMin = DelayBlockOutput
if FracMin >= 1.0 then Vmin = Vloff
else if FracMin <= 0.0 then Vmin = V2off
else
Vmin = V2off + FracMin * (Vloff - V2off)
end
```

Following Function for calculating a new DelayBlockInput

```
if PresentV <= Vmin then begin
if PresentV <= V2off then result = 0.0 // aaaa
else result = (PresentV - V2off) / (Vloff - V2off) // bbbb red curve
end
else if (Vmin >= Vloff) then result = 1.0 // cccc purple curve
else if (PresentV <= V2on) then result = FracMin // dddd light blue curve
else if (PresentV < Vlon) then begin
// see image on right above for situation when Vmin > V2on
if Vmin > V2on then tempV = Vmin
else tempV = V2on
result = FracMin + Frecon * (1.0 - FracMin) * (PresentV - tempV) / (Vlon - tempV) // eeee orange
end
else result = FracMin + Frecon * (1.0 - FracMin) // ffff green curve
```