Sizing conductors

Before determining the proper size of a conductor following the NEC requirements, you need to know:

* The conditions of use of that conductor: the ambient temperature, if it is on a rooftop, how many conductors will be in raceway, how many conductors in parallel.
* The temperature rating of the terminations/equipment, if known.
* The load nominal current and the type of load: continuous, noncontinuous or a mix of both.
* The insulation of the conductor to be used, hence its temperature rating.
* The type of metal to be used, copper or aluminum.

Let us define some variables that we will reference later. This will make it easier:

* factor1: is the product of the correction and adjustment factor of the conductor under its conditions of use and its temperature rating.
* lookup-current1: is the load current divided by factor1.
* size1: is the size of the conductor you get from table 310.15(B)(16), using lookup-current1, the temperature rating of the conductor, and the conductor metal.
* ampacity1: is the ampacity of the conductor of size1 per table 310.15(B)(16).
* corrected-amp1: ampacity1 \* factor1

The following variables apply when the temperature rating of the termination is known:

* ~~factor2: is the product of the correction and adjustment factor of the conductor under its conditions of use and the temperature rating of the terminations/equipment.~~
* ~~lookup-current2: is the load current divided by factor2.~~
* size2: is the size of the conductor you get from table 310.15(B)(16), using lookup-current1~~2~~, the temperature rating of the terminations, and the conductor metal.
* ampacity2: is the ampacity of the conductor of size1 per table 310.15(B)(16) but using the temperature rating of the terminations/equipment.

Notice how ampacity2 is defined upon size1, not upon size2!

1. **If the temperature rating of the terminations/equipment is known, you proceed as follow:**

* If your factor1 = 1.0:
  + If the temperature rating of the termination/equipment is equal or greater to the temperature rating of your conductor’s insulation, use size1, otherwise, use size2.
    - Example:

Load current=130 amps

Equipment rated for 75°C

Conductor is copper, TW (rated for 60°C).

You get:

size1=# 2/0

ampacity1=145 Amps

At 145 amps, the conductor would reach a temperature of 60°C that will not harm the terminations/equipment which is rated for 75°C.

Notice that, if your conductor were THW (rated for 75°C) you could select # 1, which is rated for 130 amps. Also notice, if your conductor were a THHW (rated for 90°C) you would need to select the size based on 75°C (temperature rating of the terminations) obtaining again a # 1 conductor.

* But if your factor1 ≠ 1:
  + If corrected-amp1 <=ampacity2, size1 is the correct size for your conductor.
  + If not, size2 is your correct conductor size.
    - Example:

Ambient temperature=102°F (Adj.Factor = 0.82)

Circuit has only 2 current-carrying conductors. (Corr.Factor = 1.0)

Conductor is aluminum, TW.

Load current=210 amps.

The terminations rated for 75°C.

You get:

factor1=0.82 (0.82x1.0)

lookup-current1=210/0.82=256.1amps

size1=500 kcmil

ampacity1=260 (obtained from the 60°C column)

corrected-amp1=260\*0.82 = 213.2 amps. (At this ampacity, the conductor would reach 60°C under the conditions of use)

(now, using the temperature rating of the terminations, 75°C, for the 500 kcmil conductor)

ampacity2=310

This is the ampacity of the 500 kcmil conductor from the 75°C column.

Since corrected-amp1 <= ampacity2 (213.2<=310), size1 is the correct size, that is (500 kcmil). This is because, it is guaranteed that, for this load and these conditions of use, the conductor will never reach the temperature of terminations. If fact, we need about 310\*0.82=254.2 amps for this conductor to reach 75°C, but it will never happen, the load is drawing only 210 amps.

* + - Example 2:

Same as before except:

Conductor is THHW (90°C) (Adj.Factor = 0.91; Corr.factor = 1.0)

You get:

factor1=0.91 (0.91x1.0)

lookup-current1=210/0.91=230.8 amps

size1=300 kcmil (90°C column, AL)

ampacity1=260 (obtained from the 90°C column)

corrected-amp1=260\*0.91 = 236.6 amps.

(now, using the temperature rating of the terminations, 75°C)

ampacity2=230

This is the ampacity of the 300 kcmil conductor from the 75°C column.

Since corrected-amp1 > ampacity2 (236.6>230) you cannot use size1 (300 kcmil).

Why?

Because your conductor reaches 75°C at 230 amps, and for these conditions of use, it will reach 75°C at 230x0.91=209.3 amps, which means that at 210 amps it will be higher than 75°C, which goes over the temperature of the terminations.

The proper size is obtained from:

~~factor2=0.88 (Adj.Factor for 75°C)~~

~~lookup-current2=210/0.88=238.6~~

size2=300 kcmil (from the 75°C column)

The correct size is 300 kcmil.

Why?

~~The 300 kcmil reaches 75°C at 250 amps, but under these conditions of use, it will reach that temperature at 250x0.88=220 amps and since the load is only 210 amps the conductor and the terminations will never go over 75°C. Keep in mind that, the 75°C at 220 amps under the given conditions of use is true for whatever insulation the conductor is rated for.~~

1. **Now, if the temperature rating of the terminations/equipment is not known, you proceed as follow:**

* If load current<=100 amps:
  + t-rating=60°C
* else: (load current>100 amps)
  + if the conductor is rated for 60°C:
    - t-rating=60°C
  + else:
    - t-rating=75°C
* factor=the product of the correction and adjustment factors, obtained for t-rating.
* lookup-current=load current/factor
* size=size of the conductor obtained from table 310.15(B)(16), using lookup-current and the column corresponding to the t-rating.

Notice that the temperature rating of the conductor is ignore if the load current is <=100 amps. For higher than 100 amps, if the conductor is rated for 60°C, you still need to choose it from the 60°C column, but if the conductor is rated for 75°C or 90°C you will always choose it from the 75°C column.

The described algorithm works well when the load type is NONCONTINUOUS. If the load is CONTINUOUS or MIXED, the rule 210.19(A)(1) must apply. Notice rules 215.2(A)(1) (for feeder) and 230.42(A) (for service) are like 210.19(A)(1). Also notice, the exception paragraph for each rule, dictates that if the circuit breaker being used is 100% rated, the rule is not mandatory.

The final refactored algorithm is:

Declare factor1

if ocdp.is100PercentRated

factor1 = conduitable.getFactor

else

factor1 = Math.in(1 / load.getMCAMultiplier, conduitable.getFactor)

if factor1 ==0

temperature rating of conductor not suitable for conditions od use

return null

Load\_current\_per\_set = load\_current/numberOfSets

lookup\_current1 = Load\_current\_per\_set /factor1

size1 = getSize(conduitable.metal, conduitable.tempRating, lookup\_current1)

if Size\_cannot\_handle\_this\_current(size1)

return null

if terminationTempRating != null

if terminationTempRating >= conduitable.tempRating

if Size\_cannot\_parallel(size1)

return null

return size1

//conductor temperature rating is higher than equipment temperature rating

ampacity1 = getAmpacity(size1, conduitable.metal, conduitable.tempRating)

corrected\_amp1 = ampacity1 \* factor1

ampacity2 = getAmpacity(size1, conduitable.metal, terminationTempRating)

if corrected\_amp1 <= ampacity2

if Size\_cannot\_parallel(size1)

return null

return size1

size2 = getSize(conduitable.metal, terminationTempRating, lookup\_current1)

if Size\_cannot\_handle\_this\_current(size2)

return null

if Size\_cannot\_parallel(size2)

return null

return size2

//the termination temperature rating is unknown

if Load\_current\_per\_set <= 100

t\_rating = 60

else

if conduitable.tempRating >=75

t\_rating = 75

else

t\_rating = 60

factor1 = conduitable.getFactor(t\_rating) //external factor

lookup\_current1 = Load\_current\_per\_set / factor1

size1 = getSize(conduitable.metal, t\_rating, lookup\_current1)

if Size\_cannot\_handle\_this\_current(size1)

return null

if Size\_cannot\_parallel(size1)

return null

return size1.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* notes added on 12/25/2020 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

~~There is an oscillation of the results in this algorithm.~~

~~This oscillation appears when the termination temperature is unknown in the case where the load current is higher than 100 amps.~~

~~Here is why:~~

~~The t\_rating value depends on the temperature rating of the conductor.~~

~~Then, factor1 depends on t\_rating, which defines lookup\_current1.~~

~~Size1 depends on the t\_rating and lookup\_current1.~~

~~The resulting size can be <= #1, in which case, t\_rating must be redefined as T60. If this happens, the size of the conductor must be recalculated, as it depends on t\_rating.~~

No! t\_rating must not be redefined since the code requires that the current be <= 100 amps **or** conductor size <= 1 AWG. If the resulting conductor is <= #1 it doesn’t matter since the decision was done based on the current which is the first option in the “or” condition.

Code in Java:

*public* Size getSizePerAmpacity(*boolean* forNeutral){  
 *Function*<Size, Boolean> checkError260 = (size) ->{  
 *if*(size == *null*) {  
 resultMessages.add(ERROR260);*//ampacity too high  
 return true*;  
 }  
 *return false*;  
 };  
 *Function*<Size, Boolean> checkError270 = (size) ->{  
 *if*(ConductorProperties.*compareSizes*(size, Size.AWG\_1$0) < 0 && numberOfSets > 1) {  
 resultMessages.add(ERROR270);*//paralleled conductors < 1/0  
 return true*;  
 }  
 *return false*;  
 };  
 *//messages cleanup* resultMessages.remove(WARNN220);  
 resultMessages.remove(ERROR260);  
 resultMessages.remove(ERROR270);  
 resultMessages.remove(ERROR290);  
 *Conduitable* conduitable = \_getConduitable();  
 *double* factor1;  
 *if*(ocdp.is100PercentRated())  
 factor1 = conduitable.getCompoundFactor(); *//do not account for 1.25  
 else* factor1 = Math.*min*(1 / load.getMCAMultiplier(), conduitable.getCompoundFactor());  
  
 *if*(factor1 == 0) {  
 *//temp. rating of conductor not suitable for the ambient temperature* resultMessages.add(ERROR290);  
 *return null*;  
 }  
 *double* loadCurrentPerSet = forNeutral ?  
 load.getNeutralCurrent() / numberOfSets :  
 load.getNominalCurrent() / numberOfSets;  
 *double* lookup\_current1 = loadCurrentPerSet / factor1;  
 Size size1 = ConductorProperties.*getSizeByAmperes*(  
 lookup\_current1,  
 conduitable.getMetal(),  
 conduitable.getTemperatureRating()  
 );  
 *if*(checkError260.apply(size1))  
 *return null*;  
 *if*(terminationTempRating != *null*) {  
 *//termination temperature rating is known  
 if*(terminationTempRating.getValue() >= conduitable.getTemperatureRating().getValue()) {  
 *if*(checkError270.apply(size1))  
 *return null*;  
 circuitAmpacity = ConductorProperties.*getAmpacity*(  
 size1,  
 conduitable.getMetal(),  
 conduitable.getTemperatureRating()  
 ) \* factor1;  
 *return* size1;  
 }  
 */\*conductor temperature rating is higher than equipment  
 temperature rating. Applying rule 310.15(B)\*/  
 double* ampacity1 = ConductorProperties.*getAmpacity*(  
 size1,  
 conduitable.getMetal(),  
 conduitable.getTemperatureRating()  
 );  
 *double* corrected\_amp1 = ampacity1 \* factor1;  
 *double* ampacity2 = ConductorProperties.*getAmpacity*(  
 size1,  
 conduitable.getMetal(),  
 terminationTempRating  
 );  
 *if*(corrected\_amp1 <= ampacity2){  
 *if*(checkError270.apply(size1))  
 *return null*;  
 circuitAmpacity = ampacity2;  
 *return* size1;  
 }  
 Size size2 = ConductorProperties.*getSizeByAmperes*(  
 lookup\_current1,  
 conduitable.getMetal(),  
 terminationTempRating  
 );  
 *if*(checkError260.apply(size2))  
 *return null*;  
 *if*(checkError270.apply(size2))  
 *return null*;  
 circuitAmpacity = ConductorProperties.*getAmpacity*(  
 size2,  
 conduitable.getMetal(),  
 terminationTempRating  
 );  
 *return* size2;  
 }  
 *else* {  
 */\*termination temperature rating is unknown\*/  
 /\*future:  
 implement: 110.14(C)(1)(4) motors design letter B, C or D...\*/* TempRating t\_rating;  
 *if*(loadCurrentPerSet <= 100)  
 t\_rating = TempRating.T60;  
 *else* {  
 *if*(conduitable.getTemperatureRating().getValue() >= 75)  
 t\_rating = TempRating.T75;  
 *else* t\_rating = TempRating.T60;  
 }  
 factor1 = conduitable.getCompoundFactor(t\_rating);  
 lookup\_current1 = loadCurrentPerSet / factor1;  
 size1 = ConductorProperties.*getSizeByAmperes*(  
 lookup\_current1,  
 conduitable.getMetal(),  
 t\_rating  
 );  
 *if* (checkError260.apply(size1))  
 *return null*;  
 *if*(checkError270.apply(size1))  
 *return null*;  
 circuitAmpacity = ConductorProperties.*getAmpacity*(  
 size1,  
 conduitable.getMetal(),  
 t\_rating  
 ) \* factor1;  
 *return* size1;  
 }  
}