Aenigma's Stacking Penalty Guide

All that you ever wanted to know about stacking penalties, but never dared to ask
V0.99

Contents

Chapter 1: Introduction to stacking penalties **3**
History of the stacking penalty 3
About the author 3
Chapter 2: The formula **4**
The formula 4
The derivation of the formula 4
Test results 6
Chapter 3: The formula in practice **8**
Using the stacking penalty 8
About the numbers used for input 8
Example I: Maller with Armor Explosive Hardener II and Energized Adaptive Nano Membrane II 9
Example II: Megathron with Ion Blaster Cannon I and three Tracking Computer IIs 10
Chapter 4: Conclusion **11**
Conclusion 1 11
Conclusion 2 11
Chapter 5: Frequently Asked Questions **12**

Chapter 1: Introduction to stacking penalties

As you might have noticed, some modules will say 'Penalty: Using more than one type of this module or similar modules that affect the same attribute on the ship will be penalized'. You'll see this on several modules, from sensor boosters to tracking computers and from inertia stabilizers to armor hardeners. What does that sentence mean? It means that for every module you add above the first, the effect on the attribute is diminished. A second sensor booster won't add the same bonus as the first.

For some time people have been looking to find out exactly how the stacking penalty works. There always has been a deviation from the values you find in the game. That is, until now. I was able to derive the stacking penalty formula that fits 'exactly'. More about that later.

History of the stacking penalty
Some of you might be wondering now why i include this. Hasn't there always been a stacking penalty? Well, the answer to that is no. There was a time that Eve knew no stacking penalty and life was easy, and to good to be true.

Then nerfbats were swung and in patch 1215 came the first stacking penalties:

> *Modules that were multiplying the same attributes were causing extreme results when filling up to 8 slots of the same module. Stacking results have been reduced to correct this. Modules that will be affected by this are: damage modifiers, tracking computers & enhancers, sensor dampeners & boosters, shield hardeners & amplifiers, shield boost amplifiers and armor enhancing modules.*

This was before i started playing Eve, but it looks like it's a late 2003 addition. It was not such a bad penalty, but it had some quirks, and that's why with Red Moon Rising, late 2005, the new stacking penalty was introduced. This is still the one we are playing with.

About the author
Some of you who read this might recognize my name from either Eve or BattleClinic.com. I started playing Eve in June 2004, and have been playing it ever since.

I'm also one of the admins of the BattleClinic, and I saw the site grow from a support site for Starfleet Command 3 to Freelancer, for which I made a mod (a recent Windows XP patch broke it) to Eve. And with the growth of the site, so grew the community. Pay us a visit if you haven't done that yet.

At the BattleClinic my usual job is to help with the mathematical modelling of several tools, which have gone from reasonably simple to relatively complex.

Chapter 2: The formula

What follows here can be quite mathematical, and might not be understood by everyone. I'll try to explain what happens as well as I can. Skip to chapter 3 if you can't keep up with the mathematics used.

The formula

To scare most of you, here it is, in it's rather elegant glory:

$$X_n = X_0 \cdot \prod_{i=0}^{n-1} 1 + A_i \cdot \exp\left(-\frac{i^2}{7.1289}\right)$$

With n the number of modules and rigs fitted that all affect the same attribute.
With X_0 the start value of the attribute.
With X_n the value of the affected attribute after fitting n modules.
With i being an index number.
And A_i being the bonus to the attribute (either positive or negative), and sorted so that $|A_i| \geq |A_{i+1}|$, which means that the largest bonus comes first.

Ok, what is it? It's basically a multiplication that goes like this:

$$X_n = X_0 \cdot (1 + A_0) \cdot \left(1 + A_1 \cdot \exp\left(-\frac{1}{7.1289}\right)\right) \cdot \ldots \cdot \left(1 + A_{n-1} \cdot \exp\left(-\frac{(n-1)^2}{7.1289}\right)\right)$$

Alternatively you could write:

$$X_1 = X_0 \cdot (1 + A_0)$$

$$X_2 = X_1 \cdot \left(1 + A_1 \cdot \exp\left(-\frac{1}{7.1289}\right)\right)$$

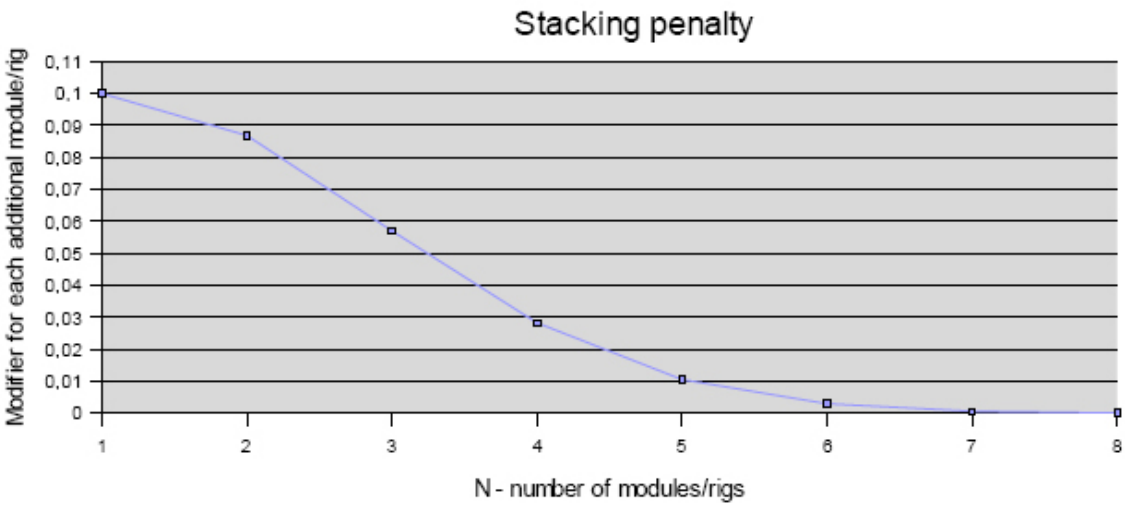
$$\vdots \qquad \vdots$$

$$X_n = X_{n-1} \cdot \left(1 + A_{n-1} \cdot \exp\left(-\frac{(n-1)^2}{7.1289}\right)\right)$$

Which is basically the way you would program it.

The derivation of the formula

The derivation was not so difficult once i found out the basic structure of the formula. If you plot each additional modifier (as i did for Heat Sink II) you get the picture you see below. As a



matter of fact, the curve describes a bell curve, which has the general formula:

$$f(x) = a \cdot \exp\left(-\frac{(x-b)^2}{c}\right)$$

with a, b, c constants. Next i made a few assumptions:

1. The first module/rig doesn't get a penalty.
2. If you could add an endless amount of modules or rigs, the last one wouldn't contribute anything, or mathematically said: $\lim_{x \to \infty} f(x) = 0$.
3. The constant a represent the attribute modifier.

From the first assumption i find that

$$f(1) = a \cdot \exp\left(-\frac{(1-b)^2}{c}\right) = a$$

and since $a \neq 0$ (because i assumed a to be the attribute modifier)

$$\exp\left(-\frac{(1-b)^2}{c}\right) = 1$$

meaning

$$(1-b)^2 = 0$$
$$b = 1$$

The second assumption proves to be true as well. I derived $c = 7.1289$ by converting the formula and just looking at the average value, using test data i gathered from Heat Sink IIs (with $a = 0.1$ for the damage modifiers).

We now have to consider if this formula works when multiplying, and it doesn't. This is because of my second assumption. What happens is that if multiplcation worked like

$$m(x) = d \cdot f(1) \cdot f(2) \cdot \ldots \cdot f(x)$$

with d a constant, the result would approach 0, even for positive bonuses, and hence positive a.

After some consideration, i made a new assumption:

4. $f(x)$ has an additive component that i managed to forget and hence i introduce

   $$g(x) = f(x) + e = a \cdot \exp\left(-\frac{(x-1)^2}{7.1289}\right) + e$$

   with $e > 0$ a constant.
5. To prevent the problem of the multiplication approaching 0, i demand $\lim_{x \to \infty} g(x) = 1$

From the second, fourth and fifth assumption i find $e = 1$. Time to look back at what we have:

$$g(x) = 1 + a \cdot \exp\left(-\frac{(x-1)^2}{7.1289}\right)$$

This should look rather familiar. Indeed, we are only three assumptions away from reaching the final result.

6. The formula should be discrete, not continuous, since the number of modules/rigs is discrete.

This last assumption is both very important, and a very logical assumption to make. Now we

convert it like this:

$$g(x) \xrightarrow{discrete} G_i = 1 + a \cdot \exp\left(-\frac{i^2}{7.1289}\right)$$

$$m(x) \xrightarrow{discrete} X_n = X_0 \cdot \prod_{i=0}^{n-1} G_i = X_0 \cdot \prod_{i=0}^{n-1}\left(1 + a \cdot \exp\left(-\frac{i^2}{7.1289}\right)\right)$$

using $X_0 = d$ and $i = 0, 1, 2, \ldots, n-1$, and of course $n = 1, 2, 3, \ldots$

Looks good, we are almost done. But what if you fit modules/rigs with different modifiers? a can't be the same then, can it? Well, logically, it won't be the same, so i assume:

7. Expanding the third assumption, we need to demand that $A_i = a$ is constant for each module/rig, but not that $A_{i+1} = A_i$ for different pairings of modules/rigs.
8. To prevent the sequence, in which you fit different modules affecting the same attribute, from having an effect on the result, and because better bonuses should not have less effect than lesser bonuses, I demand: $|A_i| \geq |A_{i+1}|$. Which means that the bonus are sorted from best bonus to least bonus.

Now we find the formula i presented at the start of the chapter:

$$X_n = X_0 \cdot \prod_{i=0}^{n-1} 1 + A_i \cdot \exp\left(-\frac{i^2}{7.1289}\right)$$

Test results

Ok, nice story, but how good is it really? I'll show you how good it is, by getting some results from Eve and comparing them to my model of the stacking penalty.

For Heat Sink IIs with $X_0 = 4.032, A_i = 0.1$

n	Damage modifier (Eve)	Damage modifier (model)	Error
1	4.43520000000	4.43520000000	0.00E+000
2	4.82067209388	4.82067209388	4.59E-012
3	5.09573151760	5.09573151760	4.03E-012
4	5.23991786724	5.23991786724	2.37E-012
5	5.29545714516	5.29545714516	1.71E-012
6	5.31133883887	5.31133883887	1.03E-012
7	5.31474350433	5.31474350433	4.36E-012
8	5.31529353802	5.31529353802	1.28E-012

For Adaptive Nano Plating I with $X_0 = 0.65, A_i = -0.08$

n	Thermal Res. (Eve)	Thermal Res. (model)	Error
1	0.598000000000	0.598000000000	0.00E+000
2	0.556421300119	0.556421300119	4.91E-013
3	0.531022530955	0.531022530956	9.49E-013

Note that here the last result shows a discrepancy in the last digit because to get values for Eve, i had to subtract the values in Eve from 1, with only 12 digits as precision. Also note that what you see in Eve might not be what's going on behind the screen. More on that later.

Chapter 2: The formula

For Inertia Stabilizer I with $X_0 = 10750000, A_i = -0.125$

n	Mass (Eve)	Mass (model)	Error
1	9406250.00000	9406250.00000	0.00E+000
2	8384355.02257	8384355.02257	4.53E-006

How to determine whether my formula is correct? Well, if you want to know for certain feel free to test every combination of modules and rigs and see if the error $|X_{n,Eve} - X_{n,model}|$ is actually smaller than the order of magnitude of the last digit. In all of the tests above this is true. I expect it to be true for everything.

Alternatively poke TomB.

Chapter 3: The formula in practice

If you bothered to read through the previous chapter, you might as well be able to calculate what I'm about to show you here. This part will be about how things stack, and how you use the formula.

Using the stacking penalty

Using the formula i can now calculate the penalty for each fitted module. Since technically you cannot fit more than 11 modules/rigs affecting the same attribute, i'll stop there. Basically the stacking penalty is given by

$$h(n) = \exp\left(-\frac{(n-1)^2}{7.1289}\right)$$

with n the number of modules.

This leads to the following table:

Number of modules/rigs	Stacking penalty
1	1.000000000000
2	0.869119980800
3	0.570583143511
4	0.282955154023
5	0.105992649743
6	0.029991166533
7	0.006410183118
8	0.001034920483
9	0.000126212683
10	0.000011626754
11	0.000000809046

As you can see, it's not very useful to use more than 3 modules that affect the same attribute, since the penalty gets pretty big (or small if you like to see it the other way) pretty fast.

A very simple way to calculate an approximate result fast (using a pocket calculator for example) is to use

$$X_1 \simeq X_0 \cdot (1 + A_0)$$
$$X_2 \simeq X_0 \cdot (1 + A_0) \cdot (1 + 0.87A_1)$$
$$X_3 \simeq X_0 \cdot (1 + A_0) \cdot (1 + 0.87A_1) \cdot (1 + 0.57A_2)$$
$$X_4 \simeq X_0 \cdot (1 + A_0) \cdot (1 + 0.87A_1) \cdot (1 + 0.57A_2) \cdot (1 + 0.28A_3)$$

Which yields the approximate results for 1, 2, 3, and 4 modules/rigs respectively. The only input are X_0 and A_0, A_1, A_2, A_3.

About the numbers used for input

X_0 is quite easy to find. Most of the times it's just the attribute the module affects and you can read it right from the screen. The situation is a bit different for shield and armor resistance. What you see on the screen is actually $1 - X_0$. If you have a kinetic resistance of 20%, that means $X_0 = 0.8$.

Chapter 3: The formula in practice

The numbers A_0, A_1, A_2, A_3 or A_i in general are a bit more complex to find, since they are sometimes misrepresented in Eve. If you recall the Damage Modifier from the Heat Sink IIs in chapter 2, you might have noticed i used $A_i = 0.1$, while ingame it says 1.1x. What i did is use the fractional increase, which is 0.1.

I made the list of the exceptions. Note that these are as they are seen ingame.

Type	Subtype	To find A_i
Electronic Warfare	Tracking Disruptor	Subtract 1
Weapon Upgrades	Gyrostabilizers	Subtract 1 from Damage Mod
Weapon Upgrades	Heat Sinks	Subtract 1 from Damage Mod
Weapon Upgrades	Magnetic Field Stabilizer	Subtract 1 from Damage Mod
Weapon Upgrades	Tracking Computers	Subtract 1
Weapon Upgrades	Tracking Enhancers	Subtract 1
Weapon Upgrades	Tracking Links	Subtract 1

The rest goes normally (keep in mind that percentages are fractions of 100, so 43% is equal to 0.43).

As noted in chapter 2, A_0, A_1, A_2, A_3 or A_i in general, are sorted so that the largest bonus comes first, and the least bonus last.

Example I: Maller with Armor Explosive Hardener II and Energized Adaptive Nano Membrane II

A pilot, who has Amarr Cruiser IV, and Explosive Armor Compensation III, wants to know what kind of explosive resistance he gets when he fits the modules mentioned above.

The Armor Explosive Hardener II has $A_i = -0.55$
The Energized Adaptive Nano Membrane II has $A_i = -0.20 \cdot 1.15 = -0.23$, taking the Explosive Armor Compensation skill into account.

The base explosive resistance of this Maller is $X_0 = 0.8 \cdot 0.8 = 0.64$, taking the resistance bonus of the Maller into account.

Since the bonus from the Hardener is larger than the one given by the Membrane, I have $A_0 = -0.55, A_1 = -0.23$

Now we can calculate the resulting resistance:

$$X_2 = 0.64 \cdot (1 - 0.55) \cdot \left(1 - 0.23 \cdot \exp\left(-\frac{1}{7.1289}\right)\right) = 0.2304294924718$$

The resistance on your screen would be 76.9570507528% and this indeed what you'll find.

Example II: Megathron with Ion Blaster Cannon I and three Tracking Computer IIs

A pilot with Gallente Battleship IV, Motion Prediction V wants to know the tracking speed of his Ion Blaster Cannons.

A Tracking Computer II has $A_i = 0.3$.

The base tracking of the Ion Blaster Cannon I is $X_0 = 0.046 \cdot 1.25 \cdot 1.3 = 0.07475$, taking both skills into account.

Since we fit three of the same module $A_0 = A_1 = A_2 = 0.3$

Now we can calculate the resulting tracking speed:

$$X_3 = 0.07475 \cdot (1 + 0.3) \cdot \left(1 + 0.3 \cdot \exp\left(-\frac{1}{7.1289}\right)\right) \cdot \left(1 + 0.3 \cdot \exp\left(-\frac{4}{7.1289}\right)\right) = 0.1434830083282$$

which, not unsurprisingly, is what you will find in game.

Chapter 4: Conclusion

I hope you enjoyed reading through this guide, or at least found it an interesting read. I'll shortly summarize the two most important conclusions you can draw from this.

Conclusion 1:
Generally, don't use more than 3 modules/rigs that affect the same attribute if they have a stacking penalty. Exception to this are of course the Remote Sensor Boosters and the Tracking Disruptors, because if you don't use them all on the same target, each target will get his own stack.

Conclusion 2:
The sequence of fitting stacking penalized modules and rigs doesn't matter. The biggest bonuses will always contribute the most.

Well that would be all there is to know about stacking penalties (and you never dared to ask).

Chapter 5: Frequently Asked Questions

Here come some frequently asked questions when it comes to stacking en stacking penalties.

Q: Do Damage Controls get a stacking penalty with armor or shield resistance modules and rigs fitted?
A: No, they don't receive a stacking penalty and apply their full bonus.

Q: Do Remote Sensor Dampeners stack with Sensor Boosters fitted on the targetted vessel?
A: As of patch 3896, the effects from Remote Sensor Dampeners are separately stacked from the effects of Sensor Boosters. The same goes for Tracking Disruptors and counter-modules like Tracking Computers.

Do you have other questions? Please let me know by sending an Eve-mail.