# PID to motor mapper

## The idea of the PID to motor mapper module

After the three PID regulators computed the values for the next orientation of the XCopter, it is necessary to take this three output values as well as the throttle value of the RCReceiver module for calculating the PWM signal for each motor. This is the task of the PIDToMotorMapper module. The motor mapper module was developed to solve two problems of mapping the input values to each motor. The first problem is, that a mathematical formula is needed which get three PID values as well as the throttle value and computes a correct PWM signal for every motor. This formula, also named mixing table or mapping table, already exists in other quadcopter projects and could be adopted. The second issue is to write a function which provides that the boost which is needed to balance out the Xcopter (the boost that is generated by the PID values) is not bigger than the boost that keeps the Xcopter in the air (the boost which is generated by the throttle value). Additionally, this function must ensure that the sum of the “throttle boost” and the “PID boost” didn´t get over 100%.

### Explanation of the mapping table

The following extract shows the mapping table of the PIDToMotorMapper module. The function PIDMIX gets a throttle value from the RCReceiver and the roll, pitch and yaw value from the PID modules. The real mapping table are the constants with the plus ones and minus ones. The function multiplies the roll, pitch and yaw value with the constants of the mapping table and sums it up with the throttle value. The return value represents the PWM signal for the motor.



### Explanation how to ensure the correct boost of the PIDs

The second problem which is described in section (ref “The idea of the PID to motor mapper module”) can be solved by limiting the throttle value to 75% and using a formula, which is described below, to compute the relation between the “throttle boost” and the “PID boost”. The advantage of limiting the throttle value to 75% is that there is a scope of +25%/-25% for the “PID boost”. A problem which can appear now, is that the “PID boost” could get a bigger value as the value of the “throttle boost”. In this situation the model couldn´t go up anymore. The following formula is used in a while loop in the motor mapper module to decrease the value of and to adjust the value of .

and are the constants which were used to adjust the and the . The two constants are both initialized with one and in case that the value of is greater than the value of , will be decremented with 0.1 and will be adjusted with the formula:

The following code snippet computes the values for the (throttleMix) and (pidMix), checks if “pidMix” is bigger than the throttle value “limitedThrottle” and returns the sum of “throttleMix” and “pidMix” which represents the value for the PWM signal. In case that “pidMix” is bigger than “limitedThrottle” the function decrements “CMix” and computes the values again.



The following function (“computeCThrottle”) computes the value for the constant .



After calculating the PWM signals for the motors the function “writeToMotors()” is called which sets the motor speed for each one. The motor speed is indicated in percent.