

НАСТРОЙКА PID РЕГУЛЯТОРОВ И GPS МОДУЛЯ ДЛЯ ПРОШИВКИ MEGAPIRATENG ПОЛЕТНОГО КОНТРОЛЛЕРА НА БАЗЕ ARDUINO MEGA2560

В работе рассматривается возможность настройки pid регуляторов и gps модуля для бюджетного БПЛА, способного выполнять полет в автоматическом режиме, который задан ему с помощью программы на компьютере. В качестве полетного контроллера используется бюджетная плата Arduino mega2560. Датчиками являются бюджетные и распространенные гироскоп-акселерометр MPU6050, компас HMC5883L, высотомер BMP180, GPS приемник u-blox NEO-6M. Рассматривается особенность использования модели pid-регулятора для обеспечения устойчивого полета квадрокоптера. Показано назначение каждого параметра pid-регулятора и описаны особенности влияния их на устойчивость полета. Рассмотрен экспериментальный подбор параметров pid-регуляторов для трех версий прошивок megariprateng для полетного контроллера. Дана практическая рекомендация настройки PID - регулятора для любых мультироторных систем. Показана возможность настройки GPS модуля u-blox NEO-6M без прошивки его flash памяти с помощью конфигурационного файла. Экспериментально установлено, что использование некорректного конфигурационного файла для прошивки GPS модуля приводит к неправильному позиционированию БПЛА, и его аварии.

Ключевые слова: PID-регулятор, GPS модуль, БПЛА, Arduino mega2560, MegapirateNG, Квадрокоптер, U-center.

Постановка задачи. В последнее время высокую степень интереса вызывают дроны или беспилотные летающие роботы, построенные на базе мультикоптеров. Среди них наиболее распространенным является четырехмоторный коптер - квадрокоптер. Сейчас они используются для исследования местности, для проведения спасательных операций МЧС, в работе пожарных служб, военной разведки. Они могут быть использованы для доставки небольших грузов на расстояние не более 5-7км со скоростью 30-40км/час. Применяются фотоаппаратами и операторами для съемок пейзажей, архитектуры, видеороликов. В настоящее время разработано большое количество полетных контроллеров с программным обеспечением. Это контроллеры Multiwii, ArduCopter (APM 2.6, APM 2.8), контроллеры DJI (Naza-M Lite, DJI Naza-M V2, DJI Wookong), MicroKopter, Zero UAV X4/X6, AutoQuad, KK., XAircraft и др. Целью работы является изучение возможности настройки pid регуляторов и

gps модуля для бюджетного БПЛА, который способен выполнять следующие основные полетные режимы:

1. Stabilize – в этом режиме выполняется взлет и посадка. Задействованы гироскоп и акселерометр для удержания горизонта.
2. AltHold – режим удержания высоты. Здесь используется барометр.
3. Land – режим автоматической посадки в текущем положении.
4. Loiter – режим удержания точки (по координате и высоте). Использует GPS.
5. RTL (Return To Launch) – возврат домой, в точку взлета.
6. Auto – полет по точкам миссии.
7. Failsafe – режим спасения, который отправляет БПЛА домой (в точку, где произведен запуск двигателей - Arming). Например, в случае потери связи с наземной станцией.

БПЛА должен быть построен на базе мультикоптера с 4-мя двигателями с рамой типоразмера F450 и расположением моторов QUAD X. В качестве прошивки изучена возможность использования MegaripateNG 2.7R4, 2.8R3 и MegaripateNG 3.1.5R2, для полетного контроллера, в состав которого входят (аналог ArduCopter):

- Микроконтроллер ATmega 2560-16AU на плате Arduino Mega2560;
- 6-ти осевой гироскоп-акселерометр MPU6050;
- 3-х осевой компас HMC5883L
- барометр/высотометр BMP180;
- GPS приемник u-blox NEO-6M.

Изложение основного материала работы. На рис. 1 показана экспериментальная модель квадрокоптера на раме типоразмера F450 и пропеллерами размером 10x45".



Рис. 1. Квадрокоптер на раме F450 с полетным контроллером на базе Arduino mega

Испытания модели показали, что для одинаковых параметрах квадрокоптера наиболее нестабильной прошивкой оказалась MegaripateNG 2.8R3. При тестовом полете наблюдались сильные осцилляции по Pitch, Roll даже при средних оборотах двигателей. Параметры PID регуляторов были установлены по умолчанию, как в прошивке. Переход к MegaripateNG 2.7R4 показал большую стабильность тестового полета. Осцилляции наблюдались лишь при высоких оборотах двигателей. Прошивка MegaripateNG 3.1.5R2 обеспечила стабильность полета квадрокоптера при любых оборотах двигателей для параметров PID регуляторов, установленных по умолчанию.

Для обеспечения надежного полета БПЛА рассмотрим схему назначения параметров PID регуляторов для обеспечения стабильного полета квадрокоптера для рассматриваемых здесь прошивках. Для лучшего представления принципа настройки параметров PID регулятора рассмотрим следующее пояснение его работы.

PID-регулятор представляет математический аппарат, который применяется почти во всех задачах стабилизации: стабилизация углов квадрокоптера в воздухе, полет и удержание позиции по GPS, удержание высоты по барометру, бесколлекторные механизмы стабилизации видеокамеры в полете (подвес камеры). Рассмотрим формулу PID-регулятора (рис. 2), предполагая, что квадрокоптер имеет только угол крена (Roll) и два двигателя (плоская модель):

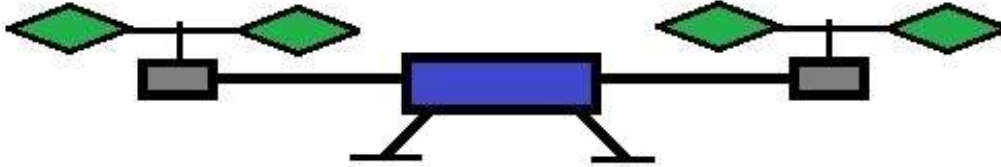


Рис. 2. Модель плоского PID-регулятора

В полетный контроллер непрерывно поступают команды с земли: «крен 30 градусов», «крен -10 градусов», «крен 0 градусов (держать горизонт)». Задача контроллера - быстрее и точнее их выполнять с помощью моторов с учетом: ветра, неравномерного распределения веса квадрокоптера, инерции квадрокоптера и т.д. Поэтому им постоянно решается задача, какую скорость вращения необходимо подавать на каждый мотор с учетом текущего значения угла крена и требуемого. Arduino одну итерацию цикла обработки и управления может выполнять за 10 миллисекунд, которые являются периодом регулирования. Чем меньше период, тем точнее происходит регулирование.

Уровень открытия дроссельной заслонки, который поступает из приемника в передатчик обозначим *throttle*. Это среднее арифметическое между скоростями вращения всех моторов, выраженное в процентах от максимальной скорости вращения. Если *left* и *right* - скорости вращения левого и правого моторов, то:

$$\begin{aligned} left &= throttle + force, \\ right &= throttle - force. \end{aligned}$$

Здесь *force* - приведенная к скорости вращения реакция (усилие) квадрокоптера, которая создает момент вращения за счет того, что левый мотор вращается на *force* быстрее, чем *throttle*, а правый на *force* медленнее. Ясно, что *force* может принимать и отрицательные значения, тогда правый мотор вращается быстрее. Для управления квадрокоптером необходимо вычислять величину *force* на каждой итерации периода регулирования. Величина *force* прежде всего должна зависеть от текущего угла крена (*roll*) и желаемого угла крена (*get_roll*) который поступает с пульта управления. Предположим, поступает команда с пульта «держать горизонт» (*get_roll=0*), а квадрокоптер имеет крен влево(рис.3).

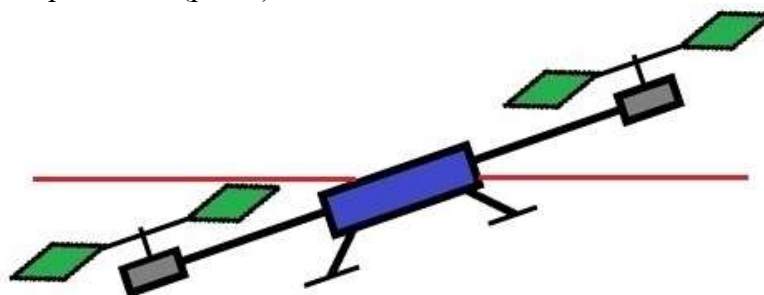


Рис. 3. Моделирование при угле крена "влево"

Пусть $error$ - разность (ошибка) между get_roll и $roll$, которую контроллер стремится минимизировать. Чем больше разность между желаемым углом крена и текущим, тем сильнее должна быть реакция, тем быстрее левый мотор должен закрутиться относительно правого. Это можно записать так:

$$force = P \cdot error . \quad (1)$$

Здесь P - коэффициент пропорциональности. Чем он больше, тем сильнее будет реакция, тем резче квадрокоптер будет реагировать на отклонение от требуемого угла крена. Эта простая формула описывает работу пропорционального регулятора. Суть ее следующая: чем сильнее квадрокоптер отклонился от требуемого положения, тем сильнее надо пытаться его вернуть. Однако эта формула должна быть усложнена вследствие перерегулирования которое можно объяснить следующим образом. За несколько десятков миллисекунд (несколько итераций цикла обработки - периодов регулирования) под воздействием пропорционального регулятора квадрокоптер вернется в требуемое (в данном случае горизонтальное) положение. Все это время ошибка $error$ и усилие $force$ будут иметь один и тот же знак, хоть и становиться все меньше по модулю. Набрав какую-то скорость поворота (угловую скорость) квадрокоптер просто перевалится на другой бок, так как формула (1) не имеет члена, приводящего к торможению. По этой причине в пропорциональный регулятор нужно добавить еще одно слагаемое, которое будет тормозить вращение квадрокоптера и препятствовать перерегулированию (переваливанию в противоположную сторону). Это слагаемое должно имитировать трение - чем быстрее поворачивается квадрокоптер, тем сильнее должна быть попытка остановить квадрокоптер. Если скорость вращения (скорость изменения ошибки) обозначить как $speed$, тогда формула регулятора запишется в виде:

$$force = P \cdot error + D \cdot speed ,$$

где D - настраиваемый коэффициент: чем он больше, тем сильнее останавливающее усилие.

Скорость изменения ошибки $speed$ можно записать так:

$$speed = derror / dt .$$

Или в разностной форме:

$$speed \sim (error - error_{previous}) / Dt .$$

Здесь: Dt - период регулирования, $error_{previous}$ - значение ошибки с предыдущей итерации цикла регуляции.

Пусть левый край будет весить чуть больше правого. Тогда квадрокоптер будет немного наклонен влево. Повернуться обратно (выровняться) он не сможет, так как дифференциальное слагаемое равно нулю, а пропорционального слагаемого не хватает, чтобы вернуть квадрокоптер в горизонтальное положение, так как левый край весит чуть больше правого. Вследствие этого квадрокоптер будет тянуть влево. Поэтому необходим механизм, который бы отслеживал такие отклонения и исправлял их. Характерной особенностью таких ошибок является то, что они проявляют себя со временем. Поэтому здесь можно ввести интегральное слагаемое. Оно хранит сумму всех ошибок $error$ по всем итерациям цикла обработки. Если после стабилизации квадрокоптера существует $error$, например 1 градус, то через 10 циклов регулирования по 0.1сек. сумма примет значение $1+0.8+0.7 + \dots < 10$ градусов. Если результат умножить на цикл регулирования, в данном примере 0.1сек, то получим число < 1 . Таким образом здесь имеем дело с интегральной величиной:

$$\Delta t \cdot \sum error .$$

Или в интегральной форме

$$\int_0^T error \cdot dt = \Delta t \cdot \sum error .$$

Окончательная формула для пропорционально-интегрально-дифференциального регулятора имеет вид:

$$force = P \cdot error + I \cdot \Delta t \cdot \sum error + D \cdot (error - error_{previous}) / \Delta t$$

или в интегральной форме:

$$force = P * error + I * \int_0^T error \cdot dt + D * \frac{derror}{dt},$$

которая совпадает с известной формулой PID - регулятора.

Ниже представлена практическая настройка PID - регулятора квадрокоптера.

P является основополагающей частью PID-регулятора, которая позволяет обеспечить хорошие летные характеристики. Практически вначале выполняют базовую настройку PID на земле при параметрах P, I, D, заданных по умолчанию. Для этого, удерживая квадрокоптер в руках:

1. Постепенно прибавляют газ, стараясь добиться невесомости коптера, когда сила тяжести уравновешивается подъёмной силой.
2. При наклонении и повороте коптера в разные стороны должна чувствоваться реакция, направленная на компенсирование прикладываемой силе;
3. Примерно на 10% каждый раз увеличивают значение коэффициента P до тех пор, пока не появится ощущение, что довольно сложно противостоять реакции коптера;
4. На последнем этапе выполняют раскачивание коптера с увеличением P до тех пор, когда начнутся осцилляции (частые вибрации) с последующим небольшим уменьшением P (до исчезновения осцилляций). Далее это выполняется для оси YAW.

Дальнейшие настройки выполняются в воздухе.

Ранее было показано, что P - это величина корректирующей силы, приложенной для того, чтобы вернуть мультикоптер в его начальное положение. Эта величина пропорциональна совокупному отклонению от изначальной позиции минус любое командное воздействие на изменение направления с пульта управления.

Более высокое значение P создаст более мощное усилие по сопротивлению любой попытке изменить положение коптера. Однако если значение P слишком велико, то при возврате в исходное положение возникает перерегулирование и, следовательно, требуется противоположная сила, чтобы компенсировать новое отклонение. Это порождает эффект раскачки до тех пор, пока наконец не будет достигнута стабильность или, в худшем случае, коптер может стать полностью неуправляемым.

Увеличение значения P приводит к большей устойчивости /стабильности до тех пор, пока слишком большое значение P не приведет к осцилляциям и потере контроля над коптером (потере управления). При большом P заметна очень большая сила воздействия, противодействующую любому изменению положения коптера.

Уменьшение значения P приведет к дрейфу в управлении. Если P слишком мало, коптер становится очень нестабильным. Коптер будет меньше сопротивляться любым попыткам изменить его положение.

Пилотажные (акробатические) полеты требуют чуть более высоких P, аккуратные и плавные полеты – чуть более низкого P.

I – это период времени, в течение которого записываются и усредняются угловые отклонения.

Величина силы, прикладываемой для возврата в исходное положение, увеличивается, если с течением времени угловое отклонение сохраняется, пока не будет достигнута максимальная величина усилия. Более высокое значение I способствует улучшению курсовой устойчивости.

Увеличение значения I улучшает способность удерживать начальное положение и уменьшает дрейф, но так же увеличивает задержку возврата в начальное положение и уменьшает влияние P.

Уменьшение значения I улучшает реакцию на изменения, но увеличивает дрейф и уменьшит способность удерживать положение. Так же увеличивает влияние P.

Пилотажный режим требует меньших значений I. Аккуратное плавное руление требует больших значений I.

D – это скорость, с которой мультикоптер вернется в начальное положение. Высокие D означают, что мультикоптер вернется в первоначальное положение очень быстро.

Увеличение значения D увеличивает скорость, с которой все отклонения будут скомпенсированы. Количество колебаний при возврате в исходное положение уменьшается. Но здесь увеличивается вероятность появления перерегулирования и осцилляций.

Уменьшение D увеличивает количество колебаний при возврате в начальное положение. Возврат в начальное положение происходит медленнее.

При пилотажном режиме увеличивают D. Аккуратные плавные полеты требуют уменьшения D.

Практическая настройка для реализации стабильных полетов:

1. Увеличивают значение P до появления осцилляций, затем его немного уменьшают;
2. Изменяют I пока коррекция после отклонения коптера не станет недопустимо слабой, затем его немного увеличивают;
3. Уменьшают D до тех пор, пока стабилизация после резких движений органами управления не станет слишком медленной. После этого D немного увеличивают.

После этих настроек можно немного уменьшить P. Должен быть выбран компромисс между оптимальными настройками стабильного зависания и обычным полетом мультикоптера. Из приведенного выше видно, что настройка PID - регулятор требует определенного опыта и искусства. Автоматическая подстройка PID параметров не представляется возможной.

Для экспериментальной модели рассматриваемого квадрокоптера на раме F450, настройки PID - регулятора представлены на рисунке 4, которые были установлены в программе mission planner для прошивки MegapirateNG 3.1.5R2.

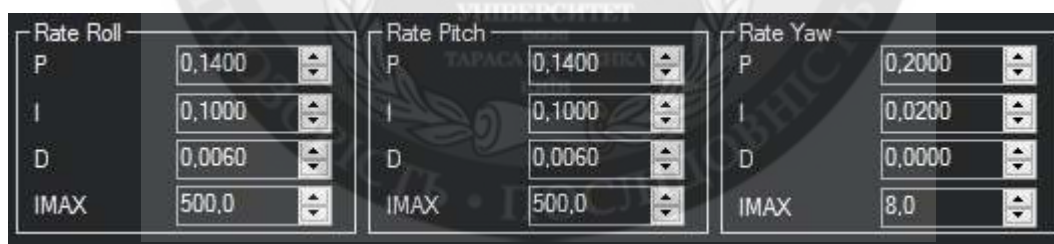


Рис.4. Значения параметров PID - регулятора в программе mission planner

Для построения полноценного беспилотного летательного аппарата (БПЛА) необходимо выполнить правильное конфигурирование GPS модуля. Благодаря этому модулю БПЛА может выполнять в автоматическом режиме полет по заданной траектории, удерживать заданную позицию, выполнять автоматический возврат в точку старта, выполнять заданные действия при достижении точки с заданными координатами и т.д. Здесь рассмотрена настройка для распространенного GPS модуля u-blox NEO-6M, который подключен к полетному контроллеру на базе Arduino Mega2560 с прошивками MegapirateNG, Ardupilot, Multiwii.



Рис. 5. GPS модуль u-blox NEO-6M с антенной, flash памятью и компасом

Для конфигурирования используется программа U-center в которой предварительно устанавливается номер и скорость (9600бит/с) порта. Далее в программе U-center (ver. 8.21) открывается вкладка Message view (View=>messages view, рис.6). При дальнейшей работе после каждого изменения параметров в окне необходимо в нижнем левом углу окна нажимать на кнопку Send (обведена красным):

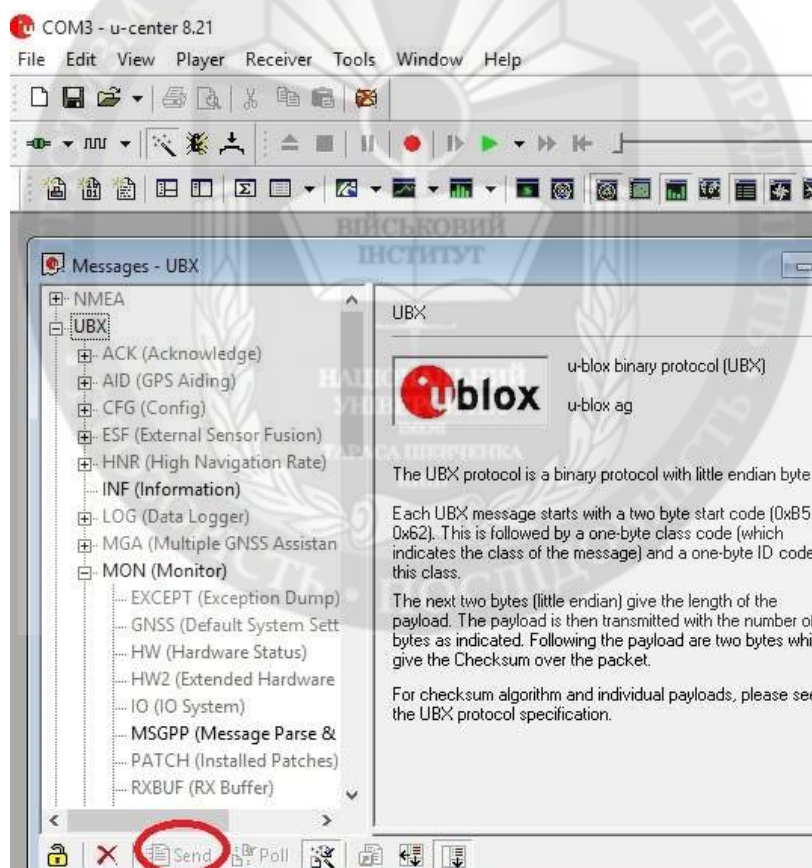


Рис. 6. Работа с программой U-center

Далее выполняются следующие действия:

1. Необходимо щелкнуть правой кнопкой мыши на слове NMEA в верхней части дерева и выбрать Disable Child Message. Слово NMEA должно поменять черный цвет на серый.
2. Выбрать Choose UBX=>CFG=>NAV5 и установить модель динамической платформы для использования: 3 - Pedestrian, а установить также Fix Mode 2 - 3D only. В этом случае

приемник GPS начнет функционировать в трехмерной системе координат. Нажать внизу на кнопку Send.

3. UBX=>CFG=>PRT – установить USART1 на скорость 38400бит/с и установить все протоколы UBX+NMEA!! Нажать внизу на кнопку Send.

4. Изменить скорость порта программы U-center на 38400бит/с. Нажать внизу на кнопку Send.

5. UBX=>CFG=>RATE(Rates) – изменить период определения координат на 200мсек. Это позволит определять позицию с частотой 5Гц, т.е. 5 раз в секунду. Нажать внизу на кнопку Send.

6. UBX=>CFG=>SBAS: Disable (SBAS перевести в состояние выключено, так как это вызывает серьезные случайные ошибки вычисления высоты). Нажать внизу на кнопку Send.

7. UBX=>NAV (но не UBX=>CFG=>NAV): здесь необходимо два раза кликнуть клавишей мыши на POSLLH, STATUS, VELNED. После этого их цвет должен измениться с серого на черный. Нажать внизу на кнопку Send.

8. UBX=>CFG=>CFG : выбрать save current config, нажать “send” в нижнем левом угле для постоянного сохранения настроек в приемнике.

9. Перейти в общем меню Receiver=> Action=> Save Config. Если этого не сделать приемник после следующего включения перейдет к первоначальным настройкам.

После этих настроек модуль GPS может быть использован на квадрокоптере и ему будет в частности доступен режим полета по путевым точкам.

Выводы. 1. Рассмотрена практическая настройка PID - регулятора на прошивках MegapirateNG 2.7R4, 2.8R3 и MegapirateNG 3.1.5R2 для полетного контроллера на основе Arduino Mega2560. Выявлен более устойчивый полет коптера на базе дефолтных настроек для прошивки MegapirateNG 3.1.5R2.

2. Показано назначение каждого параметра PID - регулятора при получении его формулы применительно для коптера в двумерном пространстве, где у него есть только один угол - угол крена.

3. Дана практическая рекомендация настройки PID - регулятора для мультироторных систем.

4. Показана возможность настройки GPS модуля u-blox NEO-6M без прошивки его flash памяти с помощью конфигурационного файла.

5. Представлено получение собственного конфигурационного файла с помощью программы U-center для прошивки GPS модуля u-blox NEO-6M.

6. Экспериментально установлено, что использование некорректного конфигурационного файла для прошивки GPS модуля приводит к неправильному позиционированию БПЛА, и может привести к его аварии.

7. Квадрокоптер, настройки которого рассмотрены здесь, по своим летным характеристикам и функционалу не уступает современным фирменным, но его стоимость примерно 4-5 раза ниже.

ЛИТЕРАТУРА:

1. Ardupilot. [Electronic resource]. - Mode of access: <http://ardupilot.org/>, 2016.
2. Полётный контроллер. - Mode of access: https://ru.wikipedia.org/wiki/Полётный_контроллер , 2016.
3. Planning a Mission with Waypoints and Events. [Electronic resource]. - Mode of access: <http://ardupilot.org/copter/docs/common-planning-a-mission-with-waypoints-and-events.html>, 2016.
4. Autopilot Hardware Options. [Electronic resource]. - Mode of access: <http://ardupilot.org/copter/docs/common-autopilots.html#common-autopilots>, 2016.
5. PID controller. [Electronic resource]. - Mode of access: https://en.wikipedia.org/wiki/PID_controller, 2017.
6. Ефимов И. И еще о ПИД - регуляторах. [Electronic resource]. - Mode of access: <http://copterpilot.ru/articles/o-pid-regulyatorax-2/>, 2014.
7. Mission Planner Home. [Electronic resource]. - Mode of access: <http://ardupilot.org/planner/>, 2017.

8. Ефимов И. Программируем квадрокоптер на Arduino (часть 1). [Electronic resource]. - Mode of access: <https://habrahabr.ru/post/227425/>, 2014.
9. U-center Windows. GNSS evaluation software for Windows. [Electronic resource]. - Mode of access: <https://www.u-blox.com/en/product/u-center-windows>, 2016.
10. Ardupirates - UbloxGPSTutorial.wiki. [Electronic resource]. - Mode of access: <https://code.google.com/archive/p/ardupirates/wikis/UbloxGPSTutorial.wiki>, 2012.
11. Планирование миссии с путевыми точками для F450 квадрокоптера с APM 2.6. [Electronic resource]. - Mode of access: <https://youtu.be/3qBPmul1Qh4>, 2016
12. НАСТРОЙКА GPS U-BLOX. [Electronic resource]. - Mode of access: <http://drone-irk.ru/?p=1404>, 2014.

REFERENCES:

1. Ardupilot. [Electronic resource]. - Mode of access: <http://ardupilot.org/>, 2016.
2. PolYotnyiy kontroller. - Mode of access: https://ru.wikipedia.org/wiki/PolYotnyiy_kontroller, 2016.
3. Planning a Mission with Waypoints and Events. [Electronic resource]. - Mode of access: <http://ardupilot.org/copter/docs/common-planning-a-mission-with-waypoints-and-events.html>, 2016.
4. Autopilot Hardware Options. [Electronic resource]. - Mode of access: <http://ardupilot.org/copter/docs/common-autopilots.html#common-autopilots>, 2016.
5. PID controller. [Electronic resource]. - Mode of access: https://en.wikipedia.org/wiki/PID_controller, 2017.
6. Efimov I. I esche o PID - regulyatorah. [Electronic resource]. - Mode of access: <http://copterilots.ru/articles/o-pid-regulyatorax-2/>, 2014.
7. Mission Planner Home. [Electronic resource]. - Mode of access: <http://ardupilot.org/planner/>, 2017.
8. Efimov I. Programmiruem kvadrokoпter na Arduino (chast 1). [Electronic resource]. - Mode of access: <https://habrahabr.ru/post/227425/>, 2014.
9. U-center Windows. GNSS evaluation software for Windows. [Electronic resource]. - Mode of access: <https://www.u-blox.com/en/product/u-center-windows>, 2016.
10. Ardupirates - UbloxGPSTutorial.wiki. [Electronic resource]. - Mode of access: <https://code.google.com/archive/p/ardupirates/wikis/UbloxGPSTutorial.wiki>, 2012.
11. Planirovanie missii s putevyimi tochkami dlya F450 kvadrokoпtera s APM 2.6. [Electronic resource]. - Mode of access: <https://youtu.be/3qBPmul1Qh4>, 2016
12. NASTROYKA GPS U-BLOX. [Electronic resource]. - Mode of access: <http://drone-irk.ru/?p=1404>, 2014.

Рецензент: д.т.н., проф. Шворов С.А., Національний університет біоресурсів і природокористування

**д.т.н., проф. Мясіщев О.А., к.т.н. Ленков Є.С., к.е.н., доц. Полозова В.М.
НАЛАШТУВАННЯ PID РЕГУЛЯТОРІВ І GPS МОДУЛЯ ДЛЯ ПРОШИВКИ MEGAPIRATENG
ПОЛЬОТНОГО КОНТРОЛЕРА НА БАЗІ ARDUINO MEGA2560**

У роботі розглядається можливість настройки pid регуляторів і gps модуля для бюджетного БПЛА, здатного виконувати політ в автоматичному режимі, який заданий йому за допомогою програми на комп'ютері. Як польотного контролера використовується бюджетна плата Arduino mega2560. Датчиками є бюджетні і поширені гіроскоп-акселерометр MPU6050, компас HMC5883L, висотомір BMP180, GPS приймач u-blox NEO-6M. Розглядається особливість використання моделі pid-регулятора для забезпечення сталого польоту квадрокоптера. Показано призначення кожного параметра pid-регулятора і описані особливості впливу їх на стійкість польоту. Розглянуто експериментальний підбір параметрів pid-регуляторів для трьох версій прошивок MegapirateNG для польотного контролера. Дана практична рекомендація налаштування PID - регулятора для будь-яких мультіроторних систем. Показана можливість настройки GPS модуля u-blox NEO-6M без прошивки його flash пам'яті за допомогою конфігураційного файлу. Експериментально встановлено, що використання некоректного конфігураційного файлу для прошивки GPS модуля призводить до неправильного позиціонування БПЛА, і його аварії.

Ключові слова: PID-регулятор, GPS модуль, БПЛА, Arduino mega2560, MegapirateNG, Квадрокоптер, U-center.

Prof. Myasishev O.A., Ph.D. Lenkov Ye.S., Ph.D. Polozova V.M.

ADJUSTMENT OF PID REGULATORS AND GPS MODULE FOR FLASHING MEGAPIRATENG FLIGHT CONTROLLER BASED ON ARDUINO MEGA2560

The paper discusses the possibility of setting up pid controllers and gps modules for a budget UAV capable of flying in automatic mode, which is assigned to it by a program on the computer. As a flight controller, the Arduino mega2560 budget board is used. Sensors are the budget and common gyro-accelerometer MPU6050, compass HMC5883L, altimeter BMP180, GPS receiver u-blox NEO-6M. The peculiarity of the use of the pid-controller model for ensuring the stable flight of a quadrocopter is considered. The purpose of each parameter of the pid-controller is shown and the features of their influence on flight stability are described. The experimental selection of the pid control parameters for the three versions of the MegapirateNG firmware for the flight controller is considered. Practical recommendation is given for tuning the PID controller for any multi-rotor systems. The possibility of setting up the u-blox NEO-6M GPS module without flashing its flash memory using the configuration file is shown. It has been experimentally established that the use of an incorrect configuration file for flashing the GPS module leads to incorrect positioning of the UAV and its failure.

Keywords: PID controller, GPS module, UAV, Arduino mega2560, MegapirateNG, Quadrocopter, U-center.