

Sikap Kestabilan Kamera Berbasis IMU dengan Metode PID

ATTITUDE STABILITY OF IMU BASED CAMERA WITH PID METHOD

Arlen Kusuma A.¹, Erwin Susanto, Ph.D.², Agung Surya Wibowo, ST., MT³

^{1,3}Prodi S1 Teknik Elektro, Fakultas Teknik Elektro, Universitas Telkom

¹arlenkusuma@students.telkomuniversity.ac.id, ²erwinelektro@telkomuniversity.ac.id,

³agungsw@telkomuniversity.ac.id

Abstrak

Dewasa ini perkembangan penerapan teknologi telah berkembang semakin pesat. Salah satu contoh video yang sedang hangat belakangan ini adalah video blog orang mengendarai motor *trail*, mobil *offroad* atau bahkan *jetski* yang diambil menggunakan sebuah *action cam*. Namun adanya perubahan pergerakan yang terjadi menyebabkan kamera tidak mampu kembali ke kondisi stabilnya.

Maka dari itu penulis melakukan sebuah penelitian dengan menggunakan sebuah sensor IMU. IMU sendiri merupakan *Inertial Measurement Unit* dimana dalam pengembangannya terdapat beberapa algoritma yang digunakan dalam pengolahan data yang akan dikeluarkan dari sensor IMU tersebut.

Metode penelitian yang digunakan adalah metode PID dan mikrokontroler yang digunakan adalah Arduino nano. Metode PID ini secara umum dapat ditemukan didalam sistem kendali seperti halnya sistem yang dirancang pada penelitian ini. Pada dasarnya metode PID terdapat tiga kendali yaitu: Proportional, Integral, dan Derivatif.

Berdasarkan hasil implementasi, kendali PI mampu memberikan persentase overshoot yang di inginkan (10%) yaitu untuk sumbu roll sebesar 9% dan sumbu pitch sebesar 12.5%

Kata kunci : Arduino nano, IMU, Motor servo, PID, Overshoot

Abstracts

Nowadays, the development of technology's implementation has grown rapidly. One of the example is a video of people riding motor trail, an offroad car, and even a jetski with the action cam that plugged on their helmet, this video has been viral recently. But, with its movement change that happened the action cam will no longer on its stable condition nor back to its stable condition.

Therefore, the writer conducted a study with the usage of IMU sensor. IMU itself is an Inertial Measurement Unit where in its development there are several algorithm that used in data processing which will be resulted from the sensor.

The method used by the writer is PID method and the microcontroller used is Arduino Nano. This method generally can be found on a control system such as the one that the writer will conduct on this study. Basically the are three controllers on PID method: Proportional, Integral, and Derivative.

Based on the result of the implementation of the writer's study, PI controller can give us the wanted overshoot (10%) which is 9% for roll axis and 12.5% for pitch axis.

Keywords : Arduino Nano, IMU, Servo motor, PID, Overshoot

Pendahuluan

Saat ini teknologi memegang peranan penting dalam kebutuhan manusia. Salah satu teknologi tersebut adalah kamera. Kamera ini bisa digunakan untuk berbagai macam kebutuhan, salah satu contoh penggunaan kamera yang akan dibahas kali ini adalah penggunaan kamera untuk keperluan *vlogging*. *Vlogging* adalah kegiatan para *vlogger* untuk mengunggah video yang telah dibuatnya ke dalam media social mereka, sama halnya dengan *blogger* tetapi para *vlogger* hanya mengunggah video saja.

"Saat ini entah sudah berapa banyak orang yang terjun menjadi *vlogger*. Sebagian adalah mereka yang memang punya *blog* dulu atau seorang *blogger*, sebagian lagi adalah mereka yang benar-benar seorang *vlogger fresh* yang tidak melalui tahapan menjadi seorang *blogger* dulu." Merupakan kutipan pembicara dari Malaysia bernama Ahmad Ihzam Omar di ajang Viral Fest Asia 2016, 15 juli 2016 yang lalu di Bali.

Melihat jumlah dari *vlogger* yang terus bertambah dan kian menjamur, kebutuhan akan kamera pun semakin meningkat. Salah satunya adalah penggunaan *action cam* atau yang saat ini lebih dikenal dengan istilah *GoPro*. Menurut hasil survey yang dilakukan Hanggara Eko Nugraha pada jum'at, 5 Februari 2016 lalu yang di unggah dalam pricebook, "Kamera aksi GoPro hadir menawarkan banyak kelebihan, utamanya adalah desain berukuran ringkas dan anti air, membuatnya fleksibel untuk dipakai dan diletakan pada posisi, baik di darat maupun di dalam air."

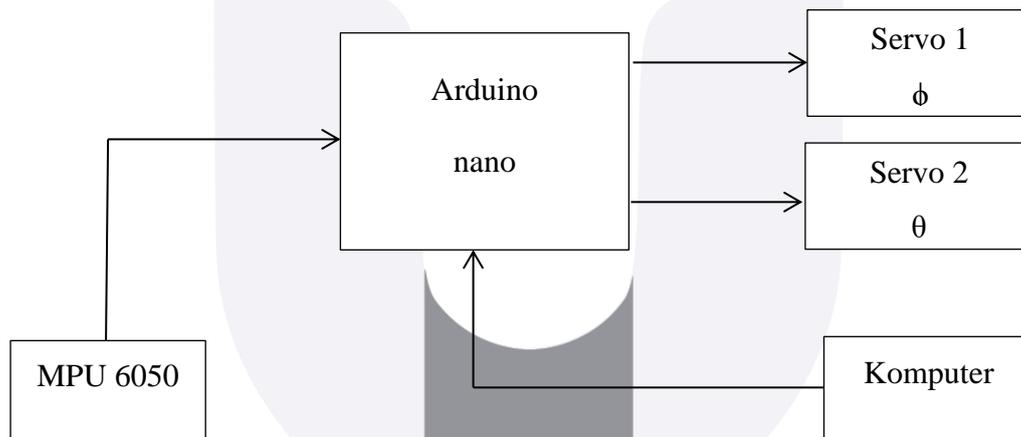
Salah satu solusi yang diteliti dari penelitian akhir ini adalah dengan merancang sebuah pengendali kestabilan dengan menggunakan mikrokontroller Arduino nano dan MPU 6050 sebagai sensornya. Pengendali kestabilan ini memudahkan *vlogger* untuk mengambil video. Atas dasar permasalahan yang akan penulis cari solusinya, maka diharapkan alat ini memiliki keunggulan yaitu dapat menstabilkan posisi kamera agar stabil di sumbu x dan y (0,0)

1. Dasar Teori

1.1. Perancangan Sistem

Dalam perancangan sistem kali ini akan dirancang alat untuk mengendalikan posisi kamera dengan menggunakan metode PID. Dalam perancangan ini akan dibagi menjadi dua bagian yaitu perancangan mekanik dan perancangan *software*. Bagian mekanik sendiri merupakan gabungan dari berbagai alat elektronik yang terdiri dari *board* Arduino nano, motor servo, dan sensor yang akan dirangkai sedemikian rupa sehingga memenuhi syarat menjadi alat yang diinginkan. Untuk bagian *software* merupakan bagian untuk pembuatan program yang akan dimasukkan ke dalam mikrokontroller yang nantinya akan berfungsi sebagai penggerak kamera.

1.1.1. Perancangan Sistem Perangkat Keras



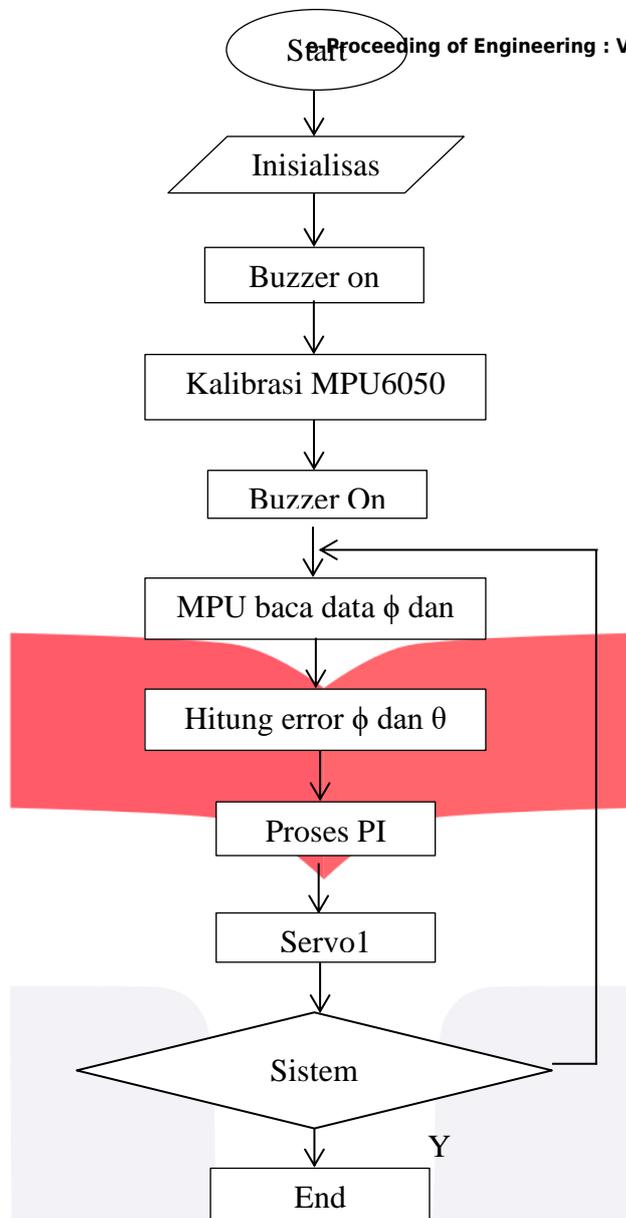
Gambar 1. Desain Perangkat Keras Sistem

motor servo 1 nantinya akan digunakan untuk daerah kerja *Roll*. Untuk motor servo 2 digunakan untuk daerah kerja *Pitch*.

Disini daerah kerja *Yaw* tidak dihubungkan dengan dengan servo karena untuk rotasi pada sumbu Y akan dilakukan manual oleh pengguna.

1.1.2. Perancangan Sistem Perangkat Lunak

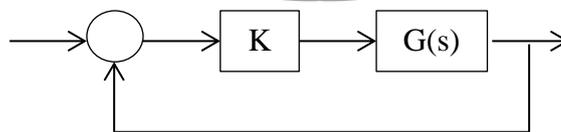
Perancangan *Software* untuk tugas akhir ini terdapat flow chart system sebagai berikut:



Gambar 2. Flow chart Sistem

1.2. Proses tuning parameter PI

Dalam proses ini, untuk dapat mengetahui parameter yang akan digunakan perlu dilakukan beberapa percobaan untuk dapat menentukan parameter PI. Penentuan parameter dilakukan dengan memakai pendekatan model matematika. Setelah didapat parameter yang dibutuhkan, langkah selanjutnya adalah memasukkan parameter tersebut ke dalam coding.



Gambar 3. blok diagram close loop

$$\frac{1}{\tau s + 1}$$

nilai τ disini adalah nilai dari suatu respon transien suatu sistem. Respon suatu sistem ini setara dengan 5 kali *time constants* atau 5τ . Sedangkan tau sendiri didapat dari servo yang bekerja. Nilai didapat dengan cara menghitung respon servo ketika diberi input dari sensor sebesar 90 derajat.

Sehingga didapatkan fungsi alih masing-masing aktuator sebagai berikut:

$$G(s) = \frac{Y(s)}{U(s)} \dots\dots\dots(1)$$

$$= \frac{1}{0.264 s + 1} ; \text{fungsi alih servo Roll}$$

$$G(s) = \frac{Y(s)}{U(s)} \dots\dots\dots(2)$$

$$= \frac{1}{0.582 s + 1} ; \text{fungsi alih servo Pitch}$$

1.3. Tuning PI untuk servo roll

Nilai ts dari persamaan dibawah adalah nilai settling time, dimana nilai tersebut akan disesuaikan dengan penelitian yang dilakukan

$$G_c(s) = \frac{\omega_n^2}{s^2 + 2 \zeta \omega_n s + \omega_n^2} \dots\dots\dots(3)$$

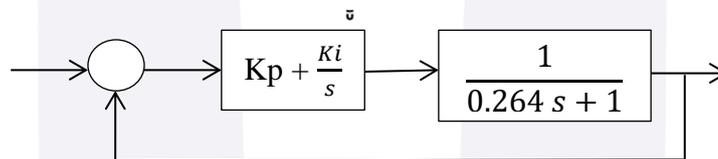
Sedangkan untuk mencari nilai ω_n dan ζ menggunakan persamaan:

$$\zeta = \frac{-\ln(\%OS/100)}{\sqrt{\pi^2 + \ln^2(\%OS/100)}} \dots\dots\dots(4)$$

$$TS = \frac{4}{\zeta \omega_n} \dots\dots\dots(5)$$

$$\omega_n = \frac{4}{\zeta TS} \dots\dots\dots(6)$$

Kemudian, langkah selanjutnya adalah menyusun fungsi alih aktuator yang didapatkan dengan kontroler PI sebagai berikut:



Gambar 4. Diagram blok close loop sumbu roll dengan kendali PI

$$TF = \frac{\frac{KpS+Ki}{0.264s^2+s}}{1 + \frac{KpS+Ki}{s} + \frac{1}{0.264s+1}} = \frac{KpS+Ki}{s(0.264s+1)+Kps+Ki} \dots\dots\dots(7)$$

$$= 0.264s^2 + (1+Kp)s + Ki \dots\dots\dots(8)$$

$$= s^2 + \frac{(1+Kp)s}{0.264} + \frac{Ki}{0.264} \dots\dots\dots(9)$$

Dengan membandingkan pole dari persamaan 3 diatas dengan persamaan 9 maka akan didapatkan nilai parameter kendali PI yang di inginkan.

1.4. Tuning PI untuk servo pitch

Nilai ts dari persamaan dibawah adalah nilai settling time, dimana nilai tersebut akan disesuaikan dengan penelitian yang dilakukan

$$G_c(s) = \frac{\omega_n^2}{s^2 + 2 \zeta \omega_n s + \omega_n^2} \dots\dots\dots(10)$$

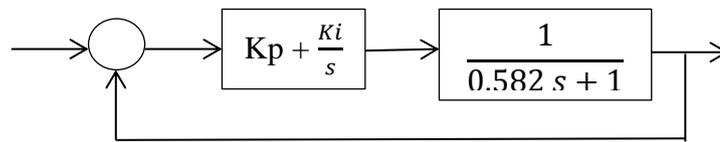
Sedangkan untuk mencari nilai ω_n dan ζ menggunakan persamaan:

$$\zeta = \frac{-\ln(\%OS/100)}{\sqrt{\pi^2 + \ln^2(\%OS/100)}} \dots\dots\dots(11)$$

$$TS = \frac{4}{\zeta\omega_n} \dots\dots\dots(12)$$

$$\omega_n = \frac{4}{\zeta TS} \dots\dots\dots(13)$$

Kemudian, langkah selanjutnya adalah menyusun fungsi alih aktuator yang didapatkan dengan kontroler PI sebagai berikut:



Gambar 5. Diagram blok close loop sumbu pitch dengan kendali PI

$$TF = \frac{\frac{KpS+Ki}{0.582S^2+S}}{1 + \frac{KpS+Ki}{S} + \frac{1}{0.582S+1}} = \frac{KpS+Ki}{s(0.582+1)+Kps+Ki} \dots\dots\dots(14)$$

$$= 0.582s^2 + (1+Kp)s + Ki \dots\dots\dots(15)$$

$$= s^2 + \frac{(1+Kp)s}{0.582} + \frac{Ki}{0.582} \dots\dots\dots(16)$$

Dengan membandingkan pole persamaan 10 diatas dengan persamaan 16 didapatkan nilai parameter kendali PI yang di inginkan

1.5. Pengujian dan Analisis

1.5.1 Pengujian sumbu roll

Tabel 2. hasil pengujian sistem *anti sway* tanpa menggunakan pengendali manual

Jenis Kendali	Settling time	Sudut	Peak data simulasi	Peak data pengujian	%OS	Keterangan
P	0.5 detik	20 ⁰	20	45	55%	Sistem masih mengalami osilasi yang besar diawal
PI	0.5 detik	30 ⁰	35	40	12.5%	<i>Overshoot</i> dan osilasi membaik namun <i>settling time</i> melambat
PI	1.5 detik	45 ⁰	50	55	9%	<i>Overshoot</i> dan osilasi membaik namun <i>settling time</i> melambat

Tabel diatas menunjukkan bahwa dengan menggunakan kendali PI mampu menurunkan overshoot sistem, namun dengan melambatkan sedikit waktu yang diperlukan sistem untuk menuju waktu tunaknya maka overshoot sistem akan semakin membaik.

1.5.2 Pengujian sumbu pitch

Tabel 3. hasil pengujian sistem *anti sway* tanpa menggunakan pengendali manual

Jenis Kontroler	Settling time	Sudut	Peak data simulasi	Peak data pengujian	%OS	Keterangan
P	1.5 detik	10 ⁰	8	22	63%	Sistem masih mengalami osilasi yang besar diawal
PI	1.5 detik	30 ⁰	35	40	12.5%	<i>Overshoot</i> dan osilasi membaik namun <i>settling time</i> melambat
PI	2 detik	25 ⁰	30	35	14.2%	osilasi sistem semakin membaik namun <i>settling time</i> melambat
PI	2.5 detik	30 ⁰	35	40	12.5%	osilasi sistem semakin membaik namun <i>settling time</i> melambat

Tabel diatas menunjukkan bahwa dengan menggunakan kendali PI mampu menurunkan overshoot sistem, namun dengan melambatkan sedikit waktu yang diperlukan sistem untuk menuju waktu tunaknya maka overshoot sistem akan semakin membaik.

2. Kesimpulan

1. Dalam hasil pengujian pada sumbu Roll didapatkan nilai parameter kendali $K_p = 0.4$ dan $K_i = 5.47$ yang bekerja dengan $t_s = 1.5$ detik agar sistem bisa bekerja dengan optimal dengan kondisi overshoot sebesar 9% atau 1% kurang dari % Overshoot yang dirancang.
2. Dalam hasil pengujian pada sumbu Pitch didapatkan nilai parameter kendali $K_p = 0.86$ dan $K_i = 4.24$ yang bekerja dengan $t_s = 2.5$ detik agar sistem bisa bekerja dengan optimal dengan kondisi overshoot sebesar 12.5% atau 2.5% lebih dari % Overshoot yang dirancang.

Daftar Pustaka :

- [1] I.J. Nagrath and M. Gopal, Control Systems Engineering 4th Edition, New Age International (P) Limited, 2006.
- [2] Karl Johan Astrom, Control System Design, Department of Mechanical and Environmental Engineering University of California, 2002.
- [3] J.C Basilio and S.R Matos, "Design of PI and PID Controllers With Transient Performance Specification", IEEE Transactions on Education. Volume 45, No.4. November 2002.
- [4] Jonathan Valdez and Jared Becker, Understanding the I2C Bus, Texas Instruments, 2015.
- [5] InvenSense.Inc, "MPU-6000 and MPU-6050 Product Specification Revision 3.4" , Agustus 2013. [online]. Available: <https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>.
- [6] Nanang Budi H. , Kemalasari , Bambang S. , Ardik W , "Makalah Pengaturan Posisi Motor Servo DC Dengan Metode P, PI, Dan PID," Politeknik Elektronika Negeri Surabaya, 2010.