

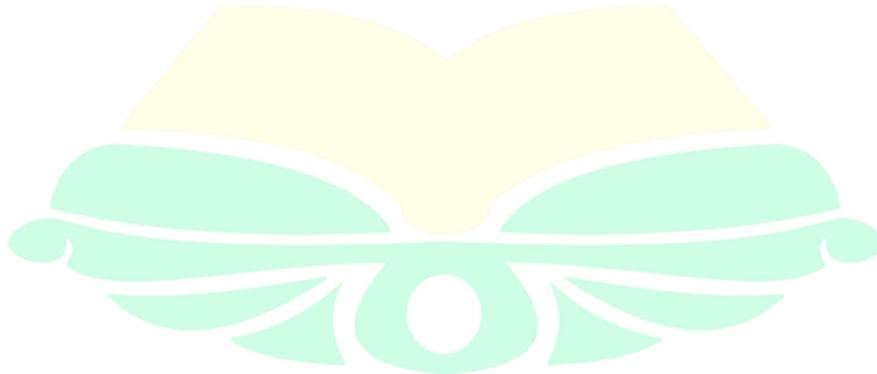
BAB 2 TINJAUAN PUSTAKA

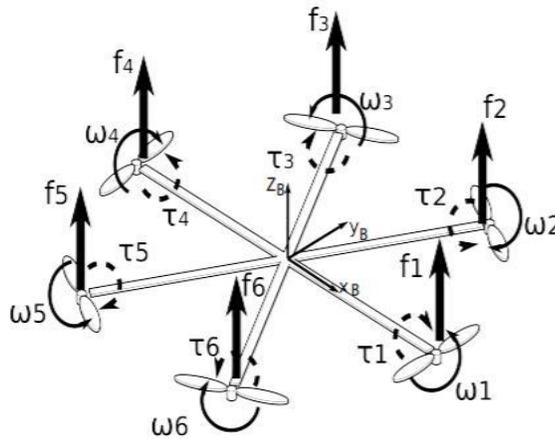
WWW.itk.ac.id

2.1 *Drone Hexacopter*

Unmanned Aerial Vehicle (UAV) memiliki beberapa jenis, salah satunya disebut *multicopter* atau *multirotor*. *Drone multicopter* adalah kendaraan terbang yang dapat memenuhi gerakan translasi dan gerakan rotasi serta diberikan variasi menurut kecepatan rotor berubah untuk mencapai gerak yang diinginkan dan untuk motor penggerak *multicopter* memiliki lebih dari dua motor (Zha dkk, 2017).

Multicopter memiliki keahlian *take-off* dan *landing* secara vertikal atau lebih dikenal menjadi VTOL (*Vertical Take-off and Landing*) dan juga *multicopter* terdiri dari dua atau lebih motor penggerak. *Design multicopter* dikembangkan dengan jumlah motor sebanyak 6 buah. *Hexacopter* mampu menahan belokan arah yang salah melalui cara menoleransi apabila ditemui kegagalan diantara motor serta mempunyai keunggulan pada membawa muatan bobot beban (*pay-load*) yang lebih besar dibandingkan dengan *quadcopter*. *Hexacopter* mempunyai konfigurasi arah putaran motor dan penempatan motor sama halnya dengan *quadcopter*, contohnya jarak yang serupa berdasarkan dari titik tengah gravitasi *hexacopter* atau biasa dianggap *center of gravity* (CG) dengan penyusunan keenam motor (Artale, 2013).





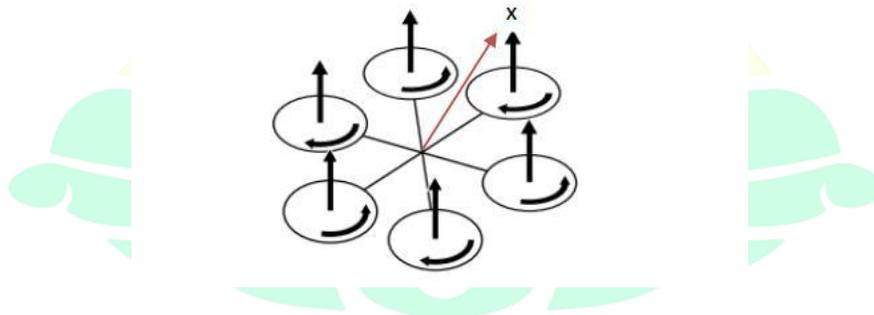
Gambar 2. 1 Bentuk Hexacopter dan Arah Putar Motor (Artale, 2013)

2.2 Pergerakan Utama pada *Hexacopter*

Hexacopter memiliki beberapa gerakan dasar yang memungkinkannya mencapai sikap (*attitude*) dan ketinggian (*altitude*) tertentu (Magnusson, 2014):

2.2.1 *Throttling/Hovering*

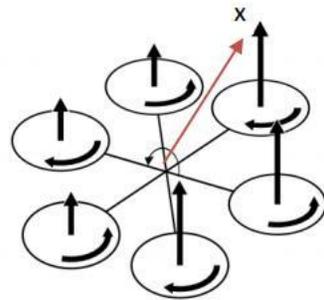
Throttling (Gambar 2.2), merupakan gerak *basic* atau pergerakan utama pada *hexacopter*. Pada saat dilakukan *throttling*, ke enam baling-baling berputar dan bergerak dengan rpm yang sama, memungkinkan pesawat bergerak vertikal ke atas dan lepas landas tanpa memerlukan banyak lahan landing.



Gambar 2. 2 Gerakan *Throttling* pada *Hexacopter* (Magnusson, 2014)

2.2.2 Rolling

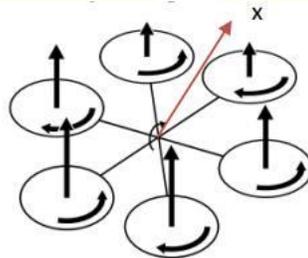
Rolling (Gambar 2.3) diperoleh dengan menambah atau mengurangi kecepatan rotasi salah satu baling-baling kiri atau kanan dan menjalankan perintah yang berlawanan pada baling-baling yang berlawanan. Gerakan menyebabkan manuver ke kiri dan kanan saat baling-baling mana yang turun dan menambah kecepatannya.



Gambar 2. 3 Gerakan *Roll* pada *Hexacopter* (Magnusson, 2014)

2.2.3 Pitching

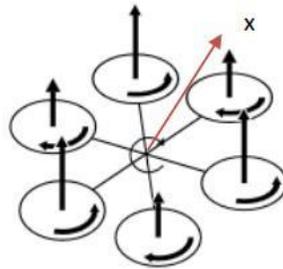
Rolling merupakan gerak baling-baling disetel ke kiri dan ke kanan. Sedangkan pada pergerakan *pitch* (Gambar 2.4), penyesuaian kecepatan baling-baling dilakukan pada baling-baling depan dan belakang. Menambah atau mengurangi kecepatan baling-baling belakang menyebabkan terjadinya manuver ke belakang/ke depan.



Gambar 2. 4 Gerakan *Pitch* pada *Hexacopter* (Magnusson, 2014)

2.2.4 Yawing

Yawing (Gambar 2.5) adalah gerak memutar *hexacopter* dengan inti pada posisi yang sama. Gerakan ini dicapai dengan memperlambat kecepatan kedua baling-baling pada putaran yang sama dan meningkatkan kecepatan putaran kedua baling-baling dengan arah yang berlawanan dari dua baling-baling sebelumnya. Ketika baling-baling kiri dan kanan melambat dan baling-baling depan dan belakang dipercepat, *hexacopter* berputar berlawanan arah jarum jam dengan inti menjadi poros. Begitu pula sebaliknya.



Gambar 2. 5 Pergerakan *Yaw* pada *Hexacopter* (Magnusson, 2014)

2.3 *Proportional Integral Derivative (PID)*

PID merupakan jaringan kendali dengan basis dasar berupa modul kendali PID, suatu kendali yang lebih sering digunakan untuk sistem kontrol industri yang kompleks. PID atau control *proportional*, *integral*, dan *derivative* adalah salah satu metode dari sistem kontrol. Metode ini merupakan kombinasi dari beberapa kontrol, seperti kontrol *proportional*, kontrol *integral* dan kontrol *derivative*. Kontrol *proportional* terutama digunakan untuk memperkuat *rise-time*. Kontrol *integral* membantu mengurangi nilai *error* dan kendali *derivative* digunakan untuk mengurangi nilai *undershot* dan *overshot*. Kombinasi ketiga kontrol ini disebut PID. Kontrol *proportional* pada dasarnya adalah penguatan yang dapat disesuaikan. Keluaran pengendali $y(t)$ dan sinyal *error* penggerak $e(t)$ (Sutanto dan Tanudjaja, 2017).

Kontrol *integral* menghasilkan respon sistem dengan memiliki *steady-state error* atau kesalahan kondisi tunak nol. Untuk sistem kontrol dengan unsur elemen *integrator* ($1/s$), kontrol *proportional* tidak dapat menjamin kinerja sistem dengan *steady-state error* atau kesalahan kondisi tunak nol. Nilai keluaran pengontrol $y(t)$ dari kontrol *integral* berubah pada kecepatan yang sebanding dengan sinyal *error* penggerak $e(t)$.

Kontrol *derivative* digunakan untuk mempercepat respon awal sistem, tetapi stabilitas *error* tidak dipertahankan. Kontrol *derivative* bekerja selama masa transisi periode, sehingga sering kendali *derivative* digunakan dalam kombinasi dengan kendali lain seperti *proportional* (Sutanto dan Tanudjaja, 2017).

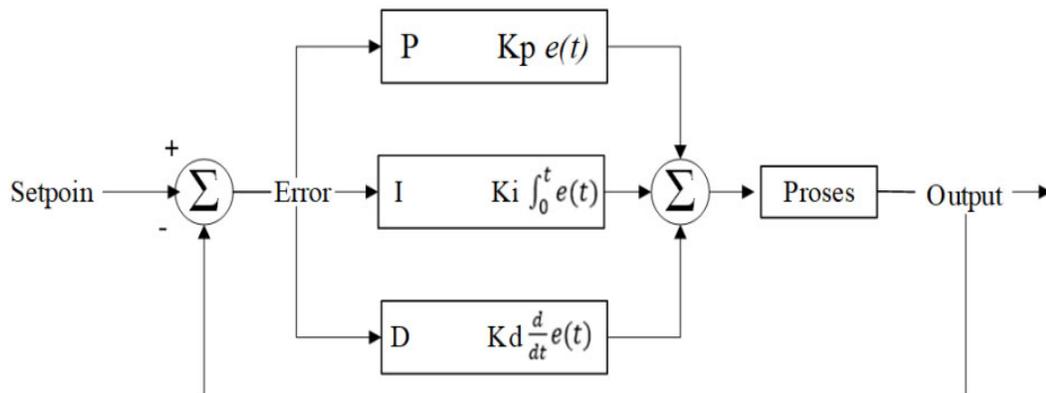
Kontroler PID memiliki kelebihan: aksi kontrol *proportional* dapat mempercepat pencapaian waktu naik (*rise time*), kontrol *integral* digunakan untuk mengurangi nilai *error* dan kontrol *derivative* digunakan untuk memperkecil nilai *undershoot/overshoot*. Tujuan digunakannya sistem kendali PID adalah untuk mendapatkan hasil kendali dengan karakteristik mengurangi nilai *error*, memperkecil *rise time*, memperkecil *settling time* dan mengurangi *overshoot* (Septiani dkk, 2017).

Dapat dilihat pada Tabel 2.1 menyajikan karakteristik masing-masing parameter sistem kontrol

Tabel 2. 1 Sifat Karakteristik Parameter Kontroler PID (Septiani dkk, 2017).

Penguatan	<i>Rise Time</i> (Rt)	<i>Overshoot</i> (Os)	<i>Settling Time</i> (St)	<i>Steady-State Error</i> (SSE)
K_p	Berkurang	Bertambah	Sedikit Berubah	Berkurang
K_i	Berkurang	Bertambah	Bertambah	Hilang
K_d	Sedikit Berubah	Berkurang	Berkurang	Sedikit Berubah

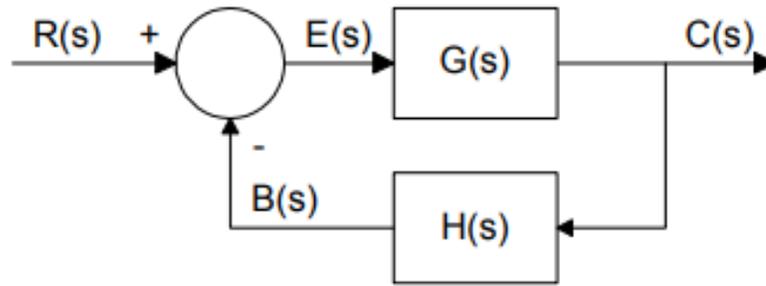
Diagram blok dari kontroler PID ditunjukkan pada Gambar 2.2 sebagai berikut



Gambar 2. 6 Diagram Blok Pengontrol PID (Maiti D. dkk, 2008)

2.4 Metode *Root Locus*

Metode *root locus* adalah metode grafis untuk mendapatkan lokasi akar dari persamaan karakteristik (kutub sistem) bila salah satu koefisien divariasikan, biasanya dari 0 hingga ∞ . Dengan metode ini, dapat menganalisis sistem dengan mengamati lokasi kutub pada nilai koefisien tertentu yang diamati. Ini memungkinkan untuk menarik kesimpulan tentang stabilitas dan respons dinamis dari sistem. Dapat juga menentukan nilai koefisien ketika kutub yang diinginkan berada di lokasi tertentu dari bidang-s. Oleh karena itu metode *root locus* dapat digunakan untuk memfasilitasi penempatan kutub. Metode *root locus* diterapkan pada *open-loop system*, dan analisisnya didasarkan pada penguatan loop daripada fungsi *closed-loop transfer*.



Gambar 2. 7 Blok Diagram Sistem Pengaturan (Haidekker, 2020)

Pertimbangan *feedback system* pada Gambar 2.7 $G(s)$ adalah sistem linear dan dapat digambarkan sebagai pecahan dari polinomial pembilang dan penyebut,

$$G(s) = \frac{p(s)}{q(s)} \quad (2.1)$$

Fungsi *closed-loop transfer* dari sistem *feedback* pada Gambar 2.7 adalah

$$H(s) = \frac{Y(s)}{X(s)} = \frac{k \frac{p(s)}{q(s)}}{1 + k \frac{p(s)}{q(s)}} \quad (2.2)$$

(Haidekker, 2020).

Root locus memiliki sifat simetri terhadap sumbu *real*, pendekatan desain dengan menggunakan metode *root locus* sangat cocok diterapkan untuk memperoleh hasil secara cepat. Metode *root locus* sangat memudahkan pengamatan pengaruh variasi suatu parameter *gain* (k) terhadap letak *pole-pole*. *Root locus* bermula dari *pole-pole* (untuk $k = 0$) dan berakhir di *zero-zero* (untuk $k = \infty$). Mencari akar-akar persamaan karakteristik untuk orde tinggi sulit dilakukan, terlebih dengan nilai *gain* (k) sebagai variabel, maka sebagai alternatif dapat dilakukan dengan menggunakan *software MATLAB* (Arifin, 2020). Metode *root locus* sendiri memiliki aturan dasar untuk membuat sketsa *root locus*, yang pertama adalah tentang simetri, bahwa *root locus* berbentuk simetris terhadap *real axis* (σ)/sumbu horizontal. Aturan nomor dua adalah jumlah dari cabang. Aturannya menyatakan bahwa jumlah cabang di sketsa *root locus* sama dengan jumlah *closed-loop poles*. Dengan kata lain bahwa jika suatu *transfer function* memiliki satu *zero* dan satu *pole* maka hanya memiliki satu cabang

(branches) yang dimana dari *pole* menuju *zero*. Apabila pada suatu grafik tidak memiliki *zero* dan memiliki dua *pole* yang berpasangan maka akan memiliki dua cabang. Aturan yang ketiga ialah bagian dari *real axis* (σ)/sumbu horizontal di sebelah kiri sejumlah dengan *open-loop poles* dan *zeros* adalah bagian dari *root locus* atau dapat disederhanakan bagian/segments pada *real axis* (σ)/sumbu horizontal yang merupakan bagian dari *root locus* (Bavafa-Toosi, 2017).

Sudut asimtot serta pusat asimtot tempat kedudukan akar dapat ditentukan, jika fungsi *loop* terbuka memiliki nilai α nol pada titik tak hingga dengan $\alpha \geq 1$, maka tempat kedudukan akar mendekati α asimtot untuk k mendekati tak hingga.

Sudut-sudut antara asimtot dan sumbu nyata:

$$KG(s)H(s) = \frac{K(b_s s^m + b_{m-1} s^{m-1} + \dots)}{s^n + a_{n-1} s^{n-1} + \dots} = \frac{K(b_m s^{m+\dots})}{s^{m+\alpha+\dots}} \quad (2.3)$$

Berdasarkan persamaan 2.3, fungsi *loop* terbuka dapat ditulis sebagai berikut

$$\lim_{s \rightarrow \infty} KG(s)H(s) = \lim_{s \rightarrow \infty} \frac{K b_m s^m}{s^n} = \lim_{s \rightarrow \infty} \frac{K b_m}{s^\alpha} \quad (2.4)$$

maka persamaan 2.4 akan mendekati harga tertingginya dengan

$$\lim_{s \rightarrow \infty} [1 + KG(s)H(s)] = \lim_{s \rightarrow \infty} \left[\frac{K b_m}{s^\alpha} \right] = 0 \quad (2.5)$$

persamaan diatas memiliki akar akar

$$s^\alpha + K b_m = 0 \quad (2.6)$$

Atau

$$\begin{aligned} s^\alpha &= -K b_m \\ &= K b_m / k \ 180^\circ, k = \pm 1, \pm 3, \dots \end{aligned} \quad (2.7)$$

Magnitude akar-akar ini mendekati tak hingga karena anggapan bahwa s mendekati tak hingga. Sudut akar-akarnya adalah harga dasar dari sudut

$$\beta = \frac{k 180^\circ}{\alpha}, k = \pm 1, \pm 3, \dots \quad (2.8)$$

Sudut dari persamaan 2.8 adalah sudut asimtot tempat kedudukan akar, karena umumnya akar-akar tempat kedudukan mendekati sudut ini di titik s (dan juga k) mendekati tak hingga. Tabel 2.2 memberikan sudut-sudut untuk harga α kecil. Untuk $\alpha = 0$, maka tidak terdapat asimtot. Tentu besar akar-akarnya adalah $(K b_m)^{1/\alpha}$.

Dapat dilihat bahwa untuk harga α yang diketahui, sudut mula-mula adalah $180^\circ/\alpha$, dengan kenaikan sebesar $360^\circ/\alpha$.

Tabel 2. 2 Sudut-sudut asimtot

α	Sudut
0	Tidak ada
1	180°
2	$\pm 90^\circ$
3	$\pm 60^\circ, \pm 180^\circ$

Sudut asimtot juga dapat ditentukan dari persamaan berikut

$$\beta = \begin{cases} \frac{(2k+1)180^\circ}{n-m}, & \text{untuk } K > 0 \\ \frac{(2k)180^\circ}{n-m}, & \text{untuk } K < 0 \end{cases} \quad (2.9)$$

Untuk $k = 0, 1, 2, \dots, n-m-2$, ini menghasilkan banyaknya asimtot yang sama dengan $n-m$. Dari persamaan 2.9 dapat ditentukan titik pusat dari cabang-cabang sudut asimtot σ_c (*centroid*) dari suatu titik pada bidang s di sumbu *real* dengan persamaan sebagai berikut

$$\sigma_c = \frac{\sum_{i=1}^n p_i - \sum_{i=1}^m z_i}{n-m} \quad (2.10)$$

Dengan p_i adalah kutub-kutub, z_i adalah nol-nol, n adalah banyaknya kutub dan m adalah nol dari $KG(s)H(s)$ (Rosalina dan Gunawan, 2016).

2.5 Kriteria Kestabilan Routh Hurwitz

Sistem linear dengan kutub n yang memiliki fungsi alih transfer:

$$H(s) = \frac{p(s)}{q(s)} = \frac{p(s)}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0} \quad (2.11)$$

stabil jika tidak ada akar dari polinomial karakteristik $q(s)$ yang memiliki komponen real positif yaitu, jika tidak ada akar yang terletak disebelah kanan sumbu imajiner pada bidang- s . Solusi numerik untuk $q(s) = 0$ dapat ditemukan, jika solusi untuk $q(s) = 0$ tidak dapat diakses, maka kriteria kestabilan *routh hurwitz* menyediakan metode yang cepat dan kuat untuk menguji apakah ada akar yang terletak di setengah

bidang kanan. Kriteria kestabilan *routh hurwitz* tidak secara eksplisit memberikan akar, tetapi memberi tahu bahwa jumlah akar di setengah bidang kanan. Karena kriteria kestabilan *routh hurwitz* dapat diterapkan bahkan jika beberapa koefisien a_k tidak diketahui atau tidak ditentukan, metode ini dapat digunakan untuk mendapatkan hasil yang cepat dengan menentukan kisaran stabil dari koefisien yang belum diketahui. Jika mengatur koefisien $q(s)$ pada Persamaan (2.3) dalam Tabel 2.2 sebagai berikut dalam setiap koefisien a_{n-k} (b_{n-k} , c_{n-k} , dll) dihitung sebagai nol untuk $k > n$:

Tabel 2. 3 Tabel Routh Hurwitz

$$q(s) = a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0$$

s^n	a_n	a_{n-2}	a_{n-4}	...
s^{n-1}	a_{n-1}	a_{n-3}	a_{n-5}	...
s^{n-2}	b_{n-1}	b_{n-3}	b_{n-5}	...
s^{n-3}	c_{n-1}	c_{n-3}	c_{n-5}	...
s^{n-4}	d_{n-1}	d_{n-3}	d_{n-5}	...
...				
s^0	...			

Polinomial $q(s)$ tidak memiliki akar di setengah bidang kanan (sistem stabil) ketika semua koefisien di kolom pertama memiliki tanda yang sama. Sebaliknya, setiap perubahan tanda pada kolom pertama menunjukkan adanya satu kutub pada setengah bidang kanan. Terkadang, elemen bernilai nol dalam *array* memerlukan perlakuan khusus yang pertama adalah satu elemen (dan hanya satu elemen) di kolom pertama adalah nol dan menyebabkan pembagian dengan nol di salah satu koefisien tingkat tinggi. Jenis elemen bernilai nol ini biasanya, kasus $b_{n-1} = 0$ tidak memungkinkan untuk menghitung koefisien c_i . Dalam hal ini, kita dapat menetapkan $b_{n-1} = \epsilon$ dengan ϵ menjadi nilai positif yang sangat kecil. Koefisien selanjutnya dapat dihitung, dan sistem kemudian dianalisis sebagai $\epsilon \rightarrow 0^+$. Yang kedua adalah lebih dari satu elemen dalam satu baris adalah nol. Hal ini umumnya menyebabkan koefisien berikutnya menjadi tidak terdefinisi sebagai konsekuensi dari nol dibagi

dengan nol. Yang ketiga adalah akar berulang pada sumbu imajiner bidang-s menunjukkan sistem yang tidak stabil. Kriteria kestabilan *routh hurwitz* tidak mengungkapkan kestabilan ini (Haidekker, 2020).

2.6 Penelitian Terdahulu

Dapat dilihat pada Tabel 2.3 tabel penelitian sebelumnya terkait saat penelitian ini berlangsung.

Tabel 2. 4 Penelitian Terdahulu

No	Nama dan Tahun Publikasi	Judul	Hasil
1	Outeiro, Pedro dkk, 2021	<i>Multiple-model control architecture for a quadrotor with constant unknown mass and inertia</i>	Dengan menggunakan metode <i>root locus</i> untuk mengetahui metode kontrol stabil menyatakan <i>linear quadratic regulator</i> (LQR) dengan aksi integratif untuk ketinggian, dan pengontrol LQR untuk sudut <i>yaw</i> .
2	Lee, Hae In dan Yoo, Dong Wan, 2017	<i>Parameter-robust linear quadratic Gaussian technique for multi-agent slung load transportation</i>	Hasil dari simulasi <i>root locus</i> nol dari pengontrol ditempatkan didekat kutub sistem pada sumbu nyata. Perubahan parameter tidak menyebabkan perubahan besar pada lintasan PRLQG. Dan jelas bahwa sistem dengan LQG/LTR dengan mudah menjadi fase non-minimum.
3	Pandey, Dheerajkumar dan T. Uppal, 2018	<i>Controller Development for a Rotary UAV using Physics Based Non-Linear Helicopter Model</i>	Menunjukkan mode helikopter yang teridentifikasi di <i>root locus</i> untuk kecepatan maju yang berbeda. Gelombang, spiral, mode penurunan <i>pitch</i> dan <i>roll</i> dalam loop terbuka stabil tetapi mode Phugoid ada disisi kanan yang

			menunjukkan ketidakstabilan.
4	Yıldırım, Çabuk, dan Bakırcıoğlu 2019	<i>Design and trajectory control of universal drone system</i>	Pengaruh perubahan jumlah lengan dan panjang lengan dari multirotor terhadap kestabilan dengan jalur <i>tracking trajectory</i> yang sama. Perubahan Panjang lengan tidak berpengaruh terhadap kestabilan wahana.
5	Khodaverdian, Lanfermann dan Adamy, 2015	<i>Root Locus Design for the Synchronization of Multi-Agent in General Directed Networks</i>	Mengusulkan strategi desain <i>root locus</i> untuk sinkronisasi sistem SISO multi-agen dengan struktur komunikasi dalam jaringan terarah. Untuk menentukan kekuatan sinkronisasi, karena persamaan karakteristik bernilai kompleks.

