

JAWABAN NOMOR 4.B URO TEST PROGRAMMING

Nama : Aryo Wisanggeni

NIM : 13523100

Jurusan : Teknik Informatika

Jelaskan algoritma-algoritma berikut dengan detail

1. Kinematics (object detection, pose estimation, camera calibration)

Sistem koordinat dalam kinematics terdapat tiga:

- Earth (e): camera pose diestimasi berdasarkan sistem koordinat ini. Orientasi sistem koordinat ini biasanya sejajar secara vertikal.
- Camera (c): sistem koordinat yang terikat pada kamera yang bergerak. Pusat dari sistem koordinat ini adalah pada optical center dari kamera dengan z-axis sejajar dengan optical axis.
- Body (b): ini adalah sistem koordinat Inertial Movement Unit (IMU).

Sinyal gyroscope $y_{\omega,t}$ mengandung perhitungan untuk kecepatan angular $\omega_{eb,t}^b$ = dari body ke earth ($_{eb}$) yang diekspresikan dalam sistem koordinat (b).

$$y_{\omega,t} = \omega_{eb,t}^b + \delta_{\omega,t}^b + e_{\omega,t}^b$$

Sinyal accelerometer $y_{a,t}$ juga mengandung perhitungan dari kombinasi percepatan vector b_t body dan vector gravitasi g . Keduanya diekspresikan dalam sistem koordinat:

$$y_{a,t} = b_t^b - g^b + \delta_{a,t}^b + e_{a,t}^b$$

Kalibrasi Pose Sensor

Camera dan IMU mempunyai perhitungan masing-masing yang diperhitungkan di sistem koordinatnya. Orientasi dari kalibrasi ditentukan dengan modifikasi dari prosedur standar kalibrasi. Hal yang sama terjadi untuk camera pose dan dari itu arah vertikal camera frame dapat ditentukan. Dengan informasi tersebut, dapat diperhitungkan rotasi antar frame.

2. ADRC (Active Disturbance Rejection Control)

Active Disturbance Rejection Control adalah teknik kontrol-bebas model yang digunakan untuk merancang pengontrol untuk sistem dengan dinamika yang tidak diketahui dan gangguan eksternal. Pendekatan ini hanya memerlukan representasi estimasi perilaku sistem untuk merancang pengontrol yang secara efektif menangkal gangguan tanpa menyebabkan overshooting.

ADRC Blok menggunakan perkiraan model orde pertama atau orde kedua dari dinamika sistem yang diketahui beserta dinamika dan gangguan yang tidak diketahui yang dimodelkan sebagai status plant yang diperluas. Biasanya, penentuan orde ini dari respons langkah open-loop dari plant dalam rentang operasi.

- First-order approximation — $\dot{y}(t) = b_0 u(t) + f(t)$

- Second-order approximation — $y(t) = b_0 u(t) + f(t)$

Dengan

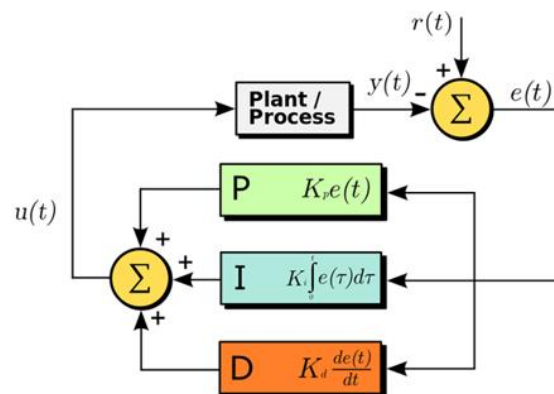
- $y(t)$ adalah plant output.
- $u(t)$ adalah input signal.
- b_0 adalah critical gain, yaitu perkiraan gain yang mendeskripsikan plant response ke suatu input $u(t)$.
- $f(t)$ adalah total disturbance yang memperhitungkan dinamika tidak diketahui dan disturbance lainnya.

Blok ini menggunakan pengamat keadaan yang diperluas (ESO) untuk memperkirakan $f(t)$ dan menerapkan pengendalian penolakan gangguan dengan mengurangi efek gangguan yang diperkirakan pada bagian perkiraan model yang diketahui.

3. PID (Proportional-Integral-Derivative) control algorithms

PID (Proportional Integral Derivative controller) merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut.

PID Blok Diagram dapat dilihat pada gambar dibawah :



Sumber: <https://imeldaazahraa.medium.com/kontrol-pid-proportional-integral-derivative-controller-c173086724af>

Adapun persamaan Pengontrol PID adalah :

$$mv(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right)$$

Keterangan :

$mv(t)$ = output dari pengontrol PID atau *Manipulated Variable*

K_p = konstanta Proporsional

T_i = konstanta Integral

T_d = konstanta Detivatif

$e(t)$ = error (selisih antara set point dengan level aktual)

Kontrol Proportional (P)

$$P\text{-out} = K_p \cdot e(t)$$

Dengan:

K_p = konstanta proportional

e = error = keadaan ideal = keadaan sekarang
t = waktu

Kontrol Integral (I)

$$I_{\text{out}} = K_i \int_0^t e(\tau) d\tau$$

KI * (Total Error + Error Sekarang)

Kontrol Derivatif (D)

KD * (Error Sekarang — Error Sebelumnya)

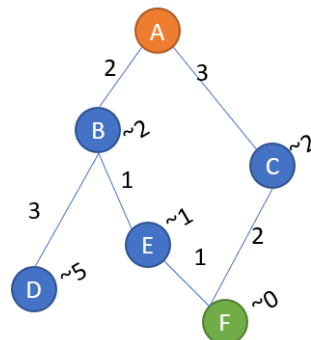
Sehingga pada akhirnya, perhitungan PID terakhir adalah

Output PID = P + I + D

4. A* (A star) algorithm

A Star adalah algoritma untuk memperkirakan jalur terpendek dalam suatu graf. Cara kerja algoritma ini adalah dengan memilih simpul berdasarkan nilai katakanlah f yang merupakan penjumlahan dua parameter lain, yaitu g dan h.

g adalah biaya untuk bergerak dari simpul awal ke simpul tertentu, sedangkan h adalah perkiraan biaya dari simpul tertentu ke tujuan akhir. Jadi dengan A Star menghitung $f = g + h$, dihitunglah secara langsung jarak dari simpul awal ke simpul tujuan untuk menentukan jalan terpendek.



Sebagai contoh adalah graf di atas, dengan A adalah awal dan F adalah tujuan.

Perhitungan B: $f = g + h = 2 + 2 = 4$

Perhitungan C: $f = g + h = 3 + 2 = 5$

Perhitungan D: $f = g + h = (2 + 3) + 5 = 10$

Perhitungan E: $f = g + h = (2 + 1) + 1 = 4$

Perhitungan F: $f = g + h = (2 + 1 + 1) + 0 = 4$

Pertama akan dicek simpul dari A, yaitu B dan C, diketahui f dari B lebih kecil dari C, sehingga B dipilih untuk diekspansi berikutnya.

Kedua akan ekspansi B ke D dan E, diketahui f dari E lebih kecil dari D, sehingga E dipilih untuk ekspansi berikutnya.

Ketiga akan ekspansi E ke F, yang ternyata sudah tujuan dan ditemukan jalan terpendeknya dengan biaya 4.