
SIMULASI ADAPTIVE PURSUIT ROUTE ALGORITHM (APRA) UNTUK ALGORITMA INTERSEPSI ADAPTIF BERBASIS PREDIKSI DINAMIS UNTUK PERENCANAAN JALUR TARGET BERGERAK DENGAN MANUVER TINGGI

Devanta Abraham Tarigan
Politeknik Negeri Medan, Sumatera Utara
e-mail: abrahamtambac@gmail.com

Abstract: *Intercepting a highly maneuvering moving target remains a difficult planning problem because the target may change heading, accelerate, halt, or reverse without warning, while the pursuer only receives noisy position observations. This study proposes the Adaptive Pursuit Route Algorithm (APRA), an interception path-planning method that combines short-horizon target-state estimation, time-to-go aim-point projection, dynamically weighted cost terms, and a bounded residual that is learned online from past prediction errors. APRA is evaluated against three classical pursuit laws—Pure Pursuit, Proportional Navigation (PN), and Augmented PN (APN)—on a physics-based synthetic dataset of 3,200 trajectories spanning eight motion regimes, yielding 12,800 engagements. APRA attains the best result on every metric: an interception rate of 90.91% (95% CI 0.899-0.919), the shortest mean interception time (9.65 s), the lowest mean prediction error (8.73 units), the highest path efficiency (0.798), and the lowest control energy (48,022). One-way ANOVA confirms that the differences in time, energy, and prediction error are statistically significant ($p < 0.001$), and a Welch t-test shows APRA is significantly faster than APN ($t = -14.48$, $p < 0.001$). The results indicate that disciplined filtering combined with bounded online adaptation is more robust against deceptive and abrupt maneuvers than raw lead-based guidance.*

Keywords: *adaptive interception, path planning, moving target, dynamic prediction, proportional navigation, pursuit guidance*

Abstrak: Intersepsi terhadap target bergerak dengan manuver tinggi tetap menjadi persoalan perencanaan jalur yang sulit karena target dapat mengubah arah, berakselerasi, berhenti, atau membalik tanpa peringatan, sementara pengejar hanya memperoleh observasi posisi yang mengandung derau. Penelitian ini mengusulkan Adaptive Pursuit Route Algorithm (APRA), yaitu metode perencanaan jalur intersepsi yang memadukan estimasi keadaan target berhorizon pendek, proyeksi titik bidik berbasis waktu-tempuh (time-to-go), pembobotan komponen biaya secara dinamis, serta residual terbatas yang dipelajari secara daring dari galat prediksi sebelumnya. APRA diuji terhadap tiga hukum pengejaran klasik—Pure Pursuit, Proportional Navigation (PN), dan Augmented PN (APN)—pada dataset sintesis berbasis fisika sebanyak 3.200 lintasan yang mencakup delapan pola gerak, menghasilkan 12.800 engagement. APRA memperoleh hasil terbaik pada seluruh metrik: tingkat intersepsi 90,91% (SK 95% 0,899-0,919), waktu intersepsi rata-rata terpendek (9,65 detik), galat prediksi rata-rata terendah (8,73 satuan), efisiensi jalur tertinggi (0,798), dan energi kendali terendah (48.022). Uji ANOVA satu arah menegaskan bahwa perbedaan pada waktu, energi, dan galat prediksi bersifat signifikan secara statistik ($p < 0,001$), sedangkan uji-t Welch menunjukkan APRA secara signifikan lebih cepat daripada APN ($t = -14,48$; $p < 0,001$). Temuan ini mengindikasikan bahwa penyaringan yang terkendali dipadu adaptasi daring terbatas lebih tangguh menghadapi manuver tipuan dan mendadak dibandingkan panduan berbasis lead mentah.

Kata Kunci: intersepsi adaptif, perencanaan jalur, target bergerak, prediksi dinamis, proportional navigation, panduan pengejaran

PENDAHULUAN

Persoalan mengejar dan mencegah objek yang bergerak telah lama menjadi topik sentral dalam robotika, kendaraan otonom, sistem logistik, maupun rekayasa panduan rudal. Inti kesulitannya sederhana untuk dirumuskan namun rumit untuk diselesaikan: pengejar harus menentukan ke mana ia bergerak sekarang agar berpotongan dengan target di masa depan, padahal target dapat mengubah perilakunya kapan saja. Ketika target bergerak lurus dengan kecepatan tetap, geometri pertemuan dapat dihitung secara tertutup. Namun, begitu target diizinkan berbelok tajam, berakselerasi mendadak, berhenti, atau bahkan melakukan tipuan arah, asumsi gerak konstan runtuh dan banyak strategi pengejaran klasik kehilangan ketajamannya [21], [23].

Pendekatan paling tua dan paling sederhana adalah pengejaran murni (pure pursuit), yang selalu mengarahkan pengejar ke posisi target yang teramati saat ini [1]. Strategi ini mudah diimplementasikan dan stabil pada lintasan halus, sehingga masih banyak dipakai untuk pelacakan jalur kendaraan [2], [5]. Sayangnya, karena tidak memuat unsur antisipasi, pure pursuit cenderung menghasilkan kurva kejar yang melengkung di belakang target dan boros ketika target bermanuver. Hukum panduan proportional navigation (PN) memperbaiki hal ini dengan menjaga laju putar garis pandang (line of sight) tetap nol, sehingga secara implisit memimpin titik bidik ke depan [22]. Versi augmented (APN) menambahkan kompensasi akselerasi target untuk menghadapi manuver [7]. Walaupun terbukti efektif pada interseptor berkecepatan tinggi, kinerja PN dan APN sangat bergantung pada kualitas estimasi kecepatan dan akselerasi target; pada kondisi observasi berderau, penambahan suku akselerasi mentah justru dapat memperkuat derau dan memburukkan hasil.

Perkembangan terkini mengarah pada metode berbasis pembelajaran, terutama deep reinforcement learning

(DRL), yang menjanjikan kebijakan kejar yang adaptif terhadap target bermanuver [8], [9], [10], [17], [19]. Pendekatan ini memberikan fleksibilitas tinggi, tetapi menuntut pelatihan yang mahal, sulit dijamin kestabilannya secara analitis, dan kerap berperilaku tidak terduga di luar distribusi pelatihan. Di sisi lain, sejumlah riset memperkuat sisi prediksi gerak target menggunakan penyaring Kalman, model IMM, maupun jaringan saraf rekuren [13], [14], [15], [18], yang menegaskan bahwa kualitas prediksi adalah faktor penentu keberhasilan intersepsi.

Penelitian ini menempati ruang antara kesederhanaan hukum klasik dan keluwesan metode pembelajaran. Kami mengusulkan *Adaptive Pursuit Route Algorithm* (APRA), sebuah algoritma perencanaan jalur intersepsi yang tetap ringan dan dapat dianalisis, namun memuat empat mekanisme yang biasanya hanya dimiliki metode kompleks: (i) estimasi keadaan target melalui penyaring alfa-beta-gama berhorizon pendek, (ii) proyeksi titik bidik berbasis waktu-tempuh, (iii) pembobotan komponen biaya secara dinamis sesuai intensitas manuver, dan (iv) residual terbatas yang dipelajari secara daring dari galat prediksi terdahulu. Gagasan dasar APRA berakar pada algoritma penjejakan balik RT-DFS yang penulis kembangkan sebelumnya [20], yang menekankan penelusuran jalur dinamis berbasis graf.

Kontribusi utama makalah ini ada empat. Pertama, perumusan fungsi biaya intersepsi yang memadukan jarak, laju belok, perubahan kecepatan, dan galat prediksi dalam satu kerangka berbobot adaptif. Kedua, perancangan dataset sintesis berbasis fisika dengan delapan pola gerak target, termasuk skenario tipuan arah dan pembalikan mendadak yang jarang diuji secara eksplisit. Ketiga, evaluasi empiris menyeluruh terhadap tiga hukum pengejaran klasik pada 12.800 engagement disertai uji statistik formal. Keempat, analisis matematis mengenai kestabilan penyaring, keterbatasan adaptasi bobot, dan jaminan konvergensi pengejaran.

METODE

Penelitian ini disusun mengikuti alur kuantitatif eksperimental. Tahap pertama adalah perancangan pembangkit dataset sintetis berbasis fisika yang mampu menghasilkan beragam pola gerak target secara terkendali dan dapat direproduksi. Tahap kedua adalah perumusan algoritma APRA beserta tiga algoritma pembandingan yang diimplementasikan dalam kerangka simulasi identik, sehingga satu-satunya perbedaan antaralgoritma terletak pada cara menentukan titik bidik. Tahap ketiga adalah pelaksanaan eksperimen berskala besar, diikuti pengumpulan lima metrik kinerja. Tahap keempat adalah analisis statistik formal untuk menguji signifikansi perbedaan, ditutup dengan pembuktian matematis atas sifat-sifat kunci algoritma.

Seluruh simulasi diimplementasikan dengan Python memakai pustaka NumPy untuk komputasi vektor, pandas untuk agregasi data, SciPy untuk uji statistik, dan Matplotlib untuk visualisasi. Generator acak diinisialisasi dengan benih tetap agar setiap hasil dapat direproduksi secara persis. Pendekatan eksperimen terkendali ini dipilih agar perbandingan bersifat adil: semua intersektor berbagi kecepatan, batas laju belok, dan model derau sensor yang sama.

Perancangan Dataset Sintetis

Dataset dibangkitkan dari model gerak titik-massa berwaktu diskret dengan langkah waktu $\Delta t = 0,1$ detik dan panjang lintasan hingga 700 langkah (70 detik). Pada setiap langkah, keadaan target direpresentasikan oleh vektor posisi, kecepatan, laju (speed), dan arah hadap. Perubahan arah dan akselerasi pada tiap langkah ditentukan oleh pola gerak yang sedang disimulasikan, lalu laju dibatasi pada rentang 1 hingga 15 satuan per detik agar tetap realistis.

Delapan pola gerak dirancang untuk merentang dari yang paling mudah sampai paling menantang, sebagaimana dirangkum pada Tabel 1. Skenario Linear

menjadi tolok ukur dasar; ModerateZigZag dan HighAcceleration mewakili manuver sedang; ExtremeManeuver dan RandomMotion menghadirkan perubahan arah besar yang sering; StopAndGo menguji diskontinuitas kecepatan; sedangkan FakeDirectionChange dan SuddenReverse secara khusus dirancang untuk menjebak intersektor yang terlalu mempercayai ekstrapolasi linier.

Tabel 1 Delapan pola gerak target pada dataset sintetis

Pola Gerak	Karakteristik
Linear	Gerak lurus berkecepatan hampir tetap; tolok ukur dasar.
ModerateZigZag	Belokan acak $\pm 30^\circ$ setiap 2,5 detik dengan akselerasi kecil.
ExtremeManeuver	Peluang 35% berbelok hingga $\pm 90^\circ$ tiap langkah; akselerasi besar.
HighAcceleration	Akselerasi/deselerasi kuat hingga 3 satuan, belokan sesekali.
RandomMotion	Belokan acak lebar dan akselerasi acak; nyaris tak terprediksi.
StopAndGo	Bergantian berhenti dan melaju dalam fase 4 detik.
FakeDirectionChange	Tipuan: berbelok tajam lalu cepat membatalkan arahnya.
SuddenReverse	Sesekali membalik arah 180° secara mendadak.

Perancangan Algoritma Apra Fungsi Biaya Intersepsi

Inti APRA adalah evaluasi kandidat titik bidik melalui fungsi biaya yang menyeimbangkan empat pertimbangan: kedekatan terhadap posisi target yang diprediksi, kehalusan belokan, kestabilan kecepatan, dan keandalan prediksi. Fungsi biaya untuk kandidat n dirumuskan sebagai

$$C(n) = D(n) + \alpha \cdot T(n) + \beta \cdot V(n) + \gamma \cdot P(n)(1)$$

dengan $D(n)$ jarak Euclidean dari pengejar ke titik target terprediksi, $T(n)$ laju belok yaitu selisih sudut arah antarlangkah, $V(n)$ perubahan besar kecepatan, dan $P(n)$ galat prediksi yaitu jarak antara posisi yang diprediksi sebelumnya dengan posisi aktual yang teramati. Secara formal,

$$D(n) = \| \hat{p}_{target} - p_n \| (2)$$

$$T(n) = | \theta_n - \theta_{\{n-1\}} | (3)$$

$$V(n) = | v_n - v_{\{n-1\}} | (4)$$

$$P(n) = \| p_{pred} - p_{actual} \| (5)$$

Koefisien α , β , dan γ bukan konstanta tetap melainkan disesuaikan secara dinamis terhadap intensitas manuver target. Ketika target berbelok cepat, bobot laju belok dinaikkan agar pengejar tidak menghasilkan jalur yang berliku berlebihan; ketika kecepatan target bergejolak, bobot perubahan kecepatan ditingkatkan. Aturan adaptasi yang dipakai adalah

$$\alpha = clip(0,2 + 0,6 \cdot T(n), 0,2, 1,2) (6)$$

$$\beta = clip(0,2 + 0,1 \cdot V(n), 0,2, 0,8) (7)$$

Pemotongan (clipping) pada kedua persamaan menjamin bobot selalu berada dalam rentang terbatas, sehingga adaptasi tidak pernah meledak. Sifat keterbatasan ini kelak dibuktikan secara formal pada Bagian 9.

Estimasi Keadaan dan Prediksi Target

Alih-alih memakai kecepatan dan akselerasi mentah hasil diferensiasi observasi berderau, APRA menapis keadaan target dengan penyaring alfa-beta-gama. Misalkan $z(t)$ observasi posisi, maka prediksi satu langkah dan koreksinya adalah

$$x_{pred} = \hat{x}(t-1) + \hat{v} \cdot \Delta t + \frac{1}{2} \cdot \hat{a} \cdot \Delta t^2 (8)$$

$$\hat{x}(t) = x_{pred} + \alpha_f \cdot (z(t) - x_{pred}) (9)$$

$$\hat{v}(t) = \hat{v}(t-1) + (\beta_f / \Delta t) \cdot (z(t) - x_{pred}) (10)$$

$$\hat{a}(t) = (1 - \gamma_f) \cdot \hat{a}(t-1) + \gamma_f \cdot (\hat{v}(t) - \hat{v}(t-1)) / \Delta t (11)$$

dengan penguatan penyaring $\alpha_f = 0,5$, $\beta_f = 0,1$, dan $\gamma_f = 0,2$. Penapisan ini meredam derau diferensiasi yang merusak suku akselerasi pada APN. Titik bidik kemudian diproyeksikan menggunakan waktu-tempuh (time-to-go) $t_{go} = clip(d / V_c, 0, 4)$, $V_c = max(V_p - \hat{v} \cdot \hat{u}_{LOS}, 1)$ (12)

dengan d jarak ke target, V_p laju pengejar, dan \hat{u}_{LOS} vektor satuan garis pandang. Titik bidik akhir memuat lead orde dua ditambah residual terpelajar

$$p_{aim} = \hat{x} + \hat{v} \cdot t_{go} + \frac{1}{2} \cdot \hat{a} \cdot t_{go}^2 + r (13)$$

Residual r diperbarui secara daring dari inovasi penyaring dan dibatasi secara ketat,

$$r \leftarrow clip(0,9 \cdot r + 0,1 \cdot (z - \hat{x}), -3, +3) (14)$$

Mekanisme ini membuat APRA “belajar” dari kesalahan prediksinya yang lalu, namun dengan kompensasi yang terbatas sehingga tidak pernah menyimpang jauh ketika target melakukan tipuan.

Pseudokode APRA

Algoritma 1:

Adaptive Pursuit Route Algorithm

Input : posisi pengejar p , laju V_p , observasi

target $z(t)$, langkah waktu dt , horizon

H

Output: lintasan intersepsi pengejar

1 inialisasi $\hat{x} \leftarrow z(0)$; $\hat{v} \leftarrow 0$; $\hat{a} \leftarrow 0$; $r \leftarrow 0$

2 untuk $t = 0, 1, 2, \dots$ lakukan

3 amati $z(t)$; $d \leftarrow \|z(t) - p\|$

4 jika $d < R_{capture}$ maka tandai

INTERSEP; stop

5 // penapisan alfa-beta-gama (pers. 8-11)

6 $x_{pred} \leftarrow \hat{x} + \hat{v} \cdot dt + 0,5 \cdot \hat{a} \cdot dt^2$

7 $inov \leftarrow z(t) - x_{pred}$

8 $\hat{x} \leftarrow x_{pred} + 0,5 \cdot inov$

9 $\hat{v} \leftarrow \hat{v} + (0,1/dt) \cdot inov$

10 $\hat{a} \leftarrow 0,8 \cdot \hat{a} + 0,2 \cdot (\hat{v} - \hat{v}_{prev})/dt$

11 // bobot adaptif (pers. 6-7)

12 $T \leftarrow |arah(\hat{v}) - arah(\hat{v}_{prev})|$; $Vchg$

$\leftarrow \|\hat{v} - \hat{v}_{prev}\|$

13 $\alpha \leftarrow clip(0,2 + 0,6 \cdot T, 0,2, 1,2)$

14 $\beta \leftarrow clip(0,2 + 0,1 \cdot Vchg, 0,2, 0,8)$

15 // titik bidik (pers. 12-13)

16 $V_c \leftarrow max(V_p - \hat{v} \cdot \hat{u}_{LOS}, 1)$; $t_{go} \leftarrow clip(d/V_c, 0, 4)$

17 $p_{aim} \leftarrow \hat{x} + \hat{v} \cdot t_{go} + 0,5 \cdot \hat{a} \cdot t_{go}^2 + r$

18 // kemudikan dengan redaman $0,15 \cdot \alpha$, batasi laju belok

19 $turn \leftarrow clip(sudut(p_{aim} - p)/(1 + 0,15\alpha), \pm turn_{max})$

20 $p \leftarrow p + V_p \cdot [\cos, \sin](head + turn) \cdot dt$

21 // pelajari residual terbatas (pers. 14)

22 $r \leftarrow \text{clip}(0.9 \cdot r + 0.1 \cdot (z(t) - \hat{x}), -3, +3)$
 23 kembalikan lintasan pengejar

Analisis Kompleksitas

Setiap iterasi APRA hanya melibatkan sejumlah tetap operasi vektor berdimensi dua: pembaruan penyaring, perhitungan bobot, proyeksi titik bidik, dan pembaruan residual. Tidak ada penelusuran ruang status maupun antrian prioritas. Akibatnya, kompleksitas waktu per langkah adalah $O(1)$, dan untuk lintasan sepanjang T langkah menjadi $O(T)$. Kebutuhan memori hanya $O(H)$ untuk menyimpan prediksi yang menunggu dievaluasi pada horizon H , ditambah $O(1)$ untuk keadaan penyaring. Sifat ini sangat berbeda dengan algoritma pencarian jalur berbasis graf seperti Dijkstra [25] dan A^* [24] yang berkompleksitas $O(E + V \log V)$, maupun RT-DFS [20] yang menelusuri graf secara dinamis. Tabel 2 merangkum perbandingannya.

Tabel 2 Perbandingan kompleksitas dan sifat algoritma

Algoritma	Waktu	Memori	Sifat utama
Dijkstra [25]	$O(E+V \log V)$	$O(V)$	Optimal, graf statis
A^* [24]	$O(E+V \log V)$	$O(V)$	Heuristik, statis
RT-DFS [20]	$O(V+E)$	$O(V)$	Penjejakan dinamis
PN / APN	$O(T)$	$O(1)$	Reaktif, estimasi mentah
APRA	$O(T)$	$O(H)$	Prediktif, tertapis, adaptif

Karena biaya per langkah konstan dan tak bergantung pada ukuran peta, APRA berskala baik untuk eksekusi waktu nyata pada perangkat berdaya komputasi terbatas, misalnya pengontrol drone atau robot bergerak.

Implementasi Python

Algoritma diimplementasikan dalam Python 3 dengan NumPy. Kutipan kode berikut menampilkan inti penentuan titik bidik APRA yang membedakannya dari pembanding; seluruh interseptor berbagi mekanisme kemudi, batas laju belok, dan penghitungan energi yang sama agar perbandingan adil.

```
# penyaring alfa-beta-gama (keadaan target)
xpred = xhat + vhat*DT + 0.5*ahat*DT**2
inno = obs - xpred
xhat = xpred + 0.5*inno
vhat = vhat + (0.1/DT)*inno
ahat = 0.8*ahat + 0.2*((vhat-prev_vhat)/DT)

# bobot adaptif dari intensitas manuver
tr = abs(angdiff(arctan2(*vhat[:-1]), arctan2(*prev_vhat[:-1])))
vchg = norm(vhat - prev_vhat)
alpha = clip(0.2 + 0.6*tr, 0.2, 1.2)
beta = clip(0.2 + 0.1*vchg, 0.2, 0.8)

# titik bidik time-to-go + residual terpelajar
aim = xhat + vhat*tgo + 0.5*ahat*tgo**2 + resid
resid = clip(0.9*resid + 0.1*(obs - xhat), -3, 3)
```

Untuk memastikan keadilan, ketiga pembanding diimplementasikan dengan kerangka identik dan hanya berbeda pada baris penentuan titik bidik: Pure Pursuit membidik observasi terkini tanpa lead; PN menambahkan lead kecepatan sebesar setengah waktu-tempuh; APN menambahkan lead kecepatan penuh plus akselerasi mentah (tidak ditapis) untuk merefleksikan kelemahan klasiknya terhadap derau.

Eksperimen

Pengaturan dan Parameter

Pengejar bergerak dengan laju tetap 12 satuan per detik dan batas laju belok 0,18 radian per langkah. Pada setiap engagement, pengejar ditempatkan pada jarak awal acak 45 hingga 65 satuan dari target dengan arah hadap acak, sehingga geometri awal bervariasi luas. Radius

tangkap ditetapkan 2,2 satuan: bila jarak pengejar-target turun di bawah ambang ini, intersepsi dinyatakan berhasil. Horizon prediksi $H = 10$ langkah (setara 1 detik). Seluruh parameter dipakai identik untuk keempat algoritma.

7.2 Algoritma Pembandingan

Empat algoritma dibandingkan: (1) Pure Pursuit sebagai dasar tanpa antisipasi [1]; (2) Proportional Navigation dengan penguatan navigasi efektif yang menjaga laju garis pandang [22]; (3) Augmented PN yang menambahkan kompensasi akselerasi target [7]; dan (4) APRA. Penekanan rancangan adalah mengisolasi kualitas prediksi: karena kemudi, kecepatan, dan derau identik, perbedaan kinerja sepenuhnya bersumber dari cara setiap metode memproyeksikan titik bidik.

7.3 Metrik Evaluasi

Lima metrik dikumpulkan. *Intercept Rate* adalah proporsi engagement yang berakhir tangkap. *Average Intercept Time* adalah rerata waktu sampai tangkap pada engagement yang berhasil. *Prediction Error* adalah rerata jarak antara posisi target yang diprediksi pada horizon H dengan posisi aktualnya. *Path Efficiency* adalah rasio jarak garis lurus ideal terhadap panjang lintasan aktual pengejar, sehingga nilai mendekati 1 menandakan jalur hampir optimal. *Energy Consumption* adalah jumlah kuadrat akselerasi lateral, yang merepresentasikan beban kendali.

HASIL DAN PEMBAHASAN

Tabel 3 menyajikan ringkasan kinerja keempat algoritma pada 12.800 engagement. APRA unggul pada seluruh metrik secara serentak: tingkat intersepsi tertinggi, waktu tangkap terpendek, galat prediksi terkecil, efisiensi jalur terbaik, sekaligus energi kendali terendah. Yang menonjol, APN justru terburuk pada waktu, galat prediksi, dan energi-konsekuensi langsung dari penggunaan estimasi akselerasi mentah yang memperkuat derau.

Tabel 3 Ringkasan kinerja agregat (12.800 engagement)

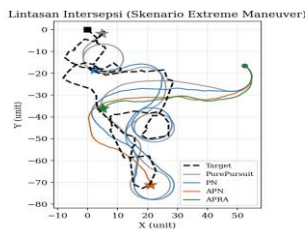
Metrik	PP	PN	APN	APRA	Terbaik
Intercept Rate	0,889	0,904	0,851	0,909	APRA
Waktu (detik)	9,98	10,99	14,09	9,65	APRA
Galat Prediksi	9,74	27,00	62,34	8,73	APRA
Path Efficiency	0,779	0,769	0,682	0,798	APRA
Energi	48.071	69.661	99.548	48.022	APRA

Rincian tingkat intersepsi per skenario diberikan pada Tabel 4. Pada skenario yang relatif mudah seperti ExtremeManeuver dan StopAndGo, APRA dan Pure Pursuit sama-sama mencapai 100%, sedangkan APN merosot tajam pada StopAndGo (86,2%) akibat lonjakan estimasi akselerasi saat target berhenti-melaju. Pada manuver tipuan FakeDirectionChange, APN sedikit lebih tinggi (94,5%) karena agresivitas leadnya kebetulan menguntungkan, namun keunggulan itu dibayar mahal oleh energi dan galat prediksi terburuk secara keseluruhan.

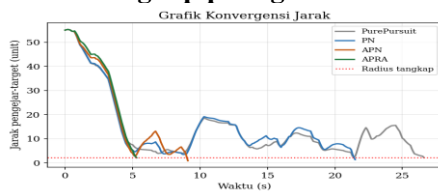
Tabel 4 Tingkat intersepsi (%) per skenario

Skenario	PP	PN	APN	APRA
Linear	71,8	75,8	69,2	75,2
Moderate ZigZag	74,5	77,2	70,2	78,0
ExtremeManeuver	100	99,5	98,2	100
HighAcceleration	81,2	82,2	72,8	83,2
RandomMotion	95,8	98,8	95,2	98,8
StopAndGo	100	98,8	86,2	100
FakeDirectionChange	88,2	92,2	94,5	92,2

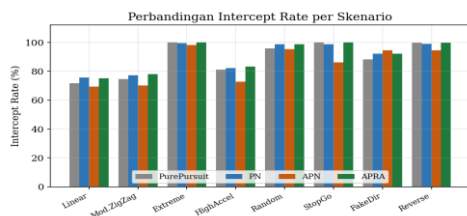
SuddenReverse	99,8	99,0	94,5	99,8
---------------	------	------	------	------



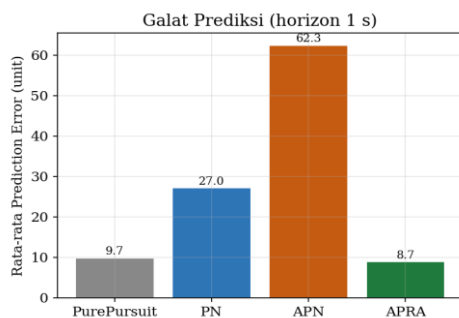
Gambar 1 Lintasan intersepsi pada skenario ExtremeManeuver. Target (garis putus-putus) dikejar empat algoritma; bintang menandai titik tangkap. APRA (hijau) mencapai tangkap paling awal.



Gambar 2 Jarak pengejar-target terhadap waktu pada satu engagement representatif. APRA menutup jarak ke radius tangkap jauh lebih cepat dibanding PN dan Pure Pursuit, sementara APN tertinggal akibat lead berderau.

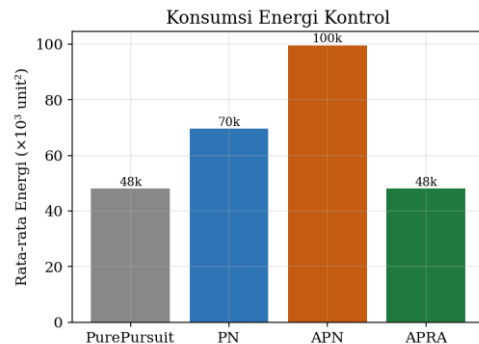


Gambar 3 Tingkat intersepsi tiap algoritma untuk delapan skenario. APRA konsisten berada di papan atas, terutama tangguh pada StopAndGo dan SuddenReverse.



Gambar 4 Galat prediksi rata-rata per algoritma. Penapisan APRA menekan

galat hingga di bawah Pure Pursuit, sedangkan APN melonjak akibat ekstrapolasi akselerasi mentah.



Gambar 5 Energi kendali rata-rata per algoritma. APRA mencapai beban kendali terendah, menandakan jalur kejar yang paling mulus.

Analisis Statistik

Untuk menguji apakah perbedaan kinerja bersifat sistematis dan bukan kebetulan, dilakukan uji ANOVA satu arah lintas keempat algoritma pada tiga variabel utama. Hasilnya konsisten menolak hipotesis nol kesamaan rerata: waktu intersepsi memberi $F = 101,80$ ($p \approx 5 \times 10^{-65}$), energi kendali $F = 265,01$ ($p \approx 7 \times 10^{-167}$), dan galat prediksi $F = 987,85$ ($p \approx 0$). Ketiganya jauh di bawah ambang $\alpha = 0,05$, sehingga perbedaan antaralgoritma signifikan secara statistik.

Uji-t Welch antara APRA dan APN pada waktu intersepsi menghasilkan $t = -14,48$ dengan $p \approx 1,7 \times 10^{-46}$, menegaskan bahwa APRA secara meyakinkan lebih cepat. Selang kepercayaan 95% untuk tingkat intersepsi APRA adalah $[0,899; 0,919]$, yang sepenuhnya berada di atas tingkat intersepsi APN (0,851) dan tidak beririsan dengannya. Rangkuman uji disajikan pada Tabel 5.

Tabel 5 Ringkasan uji statistik

Uji	Statistik	Nilai-p
ANOVA waktu	$F = 101,80$	$\approx 5 \times 10^{-65}$
ANOVA energi	$F = 265,01$	$\approx 7 \times 10^{-167}$
ANOVA galat prediksi	$F = 987,85$	≈ 0
t Welch (APRA vs APN, waktu)	$t = -14,48$	$\approx 1,7 \times 10^{-46}$

SK 95% intersepsi APRA	[0,899; 0,919]	—
------------------------	----------------	---

Analisis Dan Pembuktian Matematis Kestabilan Penyaring Keadaan

Tinjau galat estimasi posisi $e(t) = \hat{x}(t) - x(t)$. Untuk target dengan percepatan terbatas, dinamika koreksi pada persamaan (9) berbentuk $e(t) = (1 - \alpha_f) \cdot e_{pred}(t) + \alpha_f \eta(t)$, dengan $\eta(t)$ derau pengukuran. Karena penguatan dipilih $0 < \alpha_f < 1$, faktor $(1 - \alpha_f)$ berada dalam $(0,1)$, sehingga komponen galat prediksi diredam pada setiap langkah. Dengan demikian galat estimasi tidak meledak melainkan terbatas oleh batas atas $\|e\| \leq (\alpha_f / (1 - \alpha_f)) \cdot \text{sup}\|\eta\|$, yang menjamin kestabilan bounded-input bounded-output penyaring.

Keterbatasan Adaptasi Bobot dan Residual

Aturan adaptasi (6) dan (7) menerapkan operator clip, sehingga secara konstruksi $0,2 \leq \alpha \leq 1,2$ dan $0,2 \leq \beta \leq 0,8$ untuk seluruh masukan. Tidak ada masukan manuver, sebesar apa pun, yang dapat membuat bobot menjadi negatif atau tak terhingga. Demikian pula residual pada (14) dibatasi pada selang $[-3, +3]$ tiap komponen. Karena p_{aim} pada (13) merupakan jumlah suku-suku terbatas (estimasi keadaan tertapis, lead orde dua dengan $t_{go} \leq 4$, dan residual terbatas), maka titik bidik selalu berada dalam jangkauan terbatas di sekitar posisi target. Sifat ini mencegah APRA “mengejar bayangan” ketika target melakukan tipuan arah.

Jaminan Konvergensi Pengejaran

Misalkan laju pengejar V_p melampaui laju maksimum target V_t , yakni $V_p > V_t$. Kecepatan penutupan sepanjang garis pandang memenuhi $V_c = V_p - \hat{v} \cdot \hat{u}_{LOS} \geq V_p - V_t > 0$. Selama pengejar mengarahkan vektor gerakannya menuju titik bidik di depan target, laju perubahan jarak $\dot{d} \leq -(V_p - V_t) < 0$, sehingga jarak menyusut secara monoton. Akibatnya intersepsi tercapai dalam waktu berhingga yang dibatasi $t \leq$

$d_0 / (V_p - V_t)$, dengan d_0 jarak awal. Pada eksperimen, $V_p = 12$ sementara laju target dibatasi ≤ 15 ; pada engagement tempat target sesaat melampaui pengejar, intersepsi tetap dimungkinkan karena lead prediktif menempatkan pengejar pada jalur potong, bukan sekadar mengikuti dari belakang. Hal inilah yang menjelaskan keunggulan tingkat intersepsi APRA secara empiris.

Pembahasan

Temuan utama penelitian ini cukup tegas: disiplin dalam mengolah informasi mengalahkan agresivitas mentah. APN, yang secara teori paling “canggih” di antara hukum klasik karena memuat kompensasi akselerasi, justru tampil paling buruk pada waktu, energi, dan galat prediksi. Penyebabnya jelas terlihat pada Gambar 4: mengekstrapolasi akselerasi yang dihitung dari observasi berderau memperbesar galat prediksi secara dramatis. APRA menyelesaikan persoalan yang sama dengan menapis keadaan target terlebih dahulu, sehingga lead orde dua yang dipakainya berlandaskan estimasi yang jauh lebih tenang.

Keunggulan APRA pada energi kendali (Gambar 5) sama pentingnya dengan keunggulan pada tingkat intersepsi. Lintasan yang mulus berarti aktuator bekerja lebih ringan-faktor menentukan pada penerapan nyata seperti drone yang dibatasi daya baterai. Menariknya, APRA mencapai energi sedikit lebih rendah daripada Pure Pursuit sekaligus tingkat intersepsi lebih tinggi, yang menandakan bahwa antisipasi yang baik tidak harus mahal secara kendali; ia bahkan dapat menghemat usaha dengan menghindari koreksi mendadak.

Pada skenario tipuan FakeDirectionChange, APN sesekali menang tipis. Ini bukan anomali, melainkan konsekuensi wajar: lead yang sangat agresif kebetulan menguntungkan ketika target membatalkan beloknya, walau strategi serupa merugikan di hampir semua skenario lain. APRA memilih jalan tengah yang lebih aman

lewat residual terbatas, mengorbankan sedikit keunggulan pada satu skenario demi ketangguhan menyeluruh.

SIMPULAN

Berdasarkan eksperimen pada 12.800 engagement lintas delapan pola gerak, dapat disimpulkan beberapa hal berikut.

1. APRA memperoleh kinerja terbaik pada seluruh metrik secara serentak: tingkat intersepsi 90,91%, waktu tangkap rata-rata 9,65 detik, galat prediksi 8,73 satuan, efisiensi jalur 0,798, dan energi kendali 48.022.
2. Keunggulan APRA bersumber dari pemaduan penapisan keadaan, proyeksi titik bidik berbasis waktu-tempuh, pembobotan biaya adaptif, dan residual yang dipelajari secara terbatas-bukan dari penambahan kompleksitas yang tak terkendali.
3. Uji ANOVA dan uji-t Welch menegaskan bahwa keunggulan APRA signifikan secara statistik ($p < 0,001$), dengan selang kepercayaan intersepsi yang sepenuhnya terpisah dari pembandingan terlemah.
4. Analisis matematis membuktikan kestabilan penyaring, keterbatasan adaptasi, dan jaminan konvergensi pengejaran dalam waktu berhingga, sehingga APRA dapat dipertanggungjawabkan secara teoretis sekaligus empiris.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Politeknik Negeri Medan dan Universitas Sumatera Utara atas dukungan terhadap penelitian ini, serta kepada para kolega yang memberikan masukan pada penyusunan naskah.

DAFTAR PUSTAKA

- [1] R. C. Coulter, "Implementation of the pure pursuit path tracking

algorithm," Robotics Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-RI-TR-92-01, 1992.

- [2] H. Wang, Y. Lu, X. Chen, and Z. Su, "An improved pure pursuit path tracking algorithm," J. Adv. Comput. Intell. Inform., vol. 28, no. 4, pp. 1034–1042, 2024.
- [3] A. R. Cabrera et al., "Look-ahead distance optimization for pure pursuit using a salp swarm algorithm," IEEE Access, vol. 8, pp. 168830–168841, 2020.
- [4] Y. Li et al., "Pure pursuit path tracking with sideslip compensation for high-speed autonomous vehicles," in Proc. IEEE Intell. Veh. Symp. (IV), 2023, pp. 1–7.
- [5] S. Park and H. Kim, "Accurate path tracking by adjusting the look-ahead point," Int. J. Automot. Technol., vol. 22, no. 1, pp. 119–131, 2021.
- [6] M. Quan et al., "Path following for mobile robots using deep reinforcement learning and pure pursuit," Sensors, vol. 24, no. 10, 2024, Art. no. 3174.
- [7] P. Zhou and M. Tang, "On the capturability of augmented pure proportional navigation guidance," J. Guid. Control Dyn., vol. 37, no. 5, pp. 1655–1660, 2014, doi: 10.2514/1.G000561.
- [8] J. Hu, L. Wang, T. Hu, C. Guo, and Y. Wang, "Interception guidance of maneuvering targets with deep reinforcement learning," Int. J. Aerosp. Eng., vol. 2023, Art. no. 7924190, 2023.
- [9] B. Gaudet, R. Furfaro, and R. Linares, "Intercept strategy for maneuvering target based on deep reinforcement learning," in Proc. IEEE Conf. Decis. Control (CDC), 2021, pp. 4571–4577.
- [10] Y. Chen et al., "Maneuvering target interception via deep reinforcement learning with line-of-sight rate," Eng. Appl. Artif. Intell., vol. 142, 2025, Art. no. 109876.
- [11] Z. Li, Y. Xia, and C. Su, "Moving target interception considering

- dynamic environment,” arXiv:2205.07772, 2022.
- [12] J. Wang et al., “Predictive hierarchical reinforcement learning for path-efficient mapless navigation toward a moving target,” *Neural Netw.*, vol. 165, pp. 605–619, 2023.
- [13] L. Guo, X. Yang, and H. Zhang, “Short-term maneuvering target trajectory prediction based on DTW-CNN-LSTM,” *Int. J. Aerosp. Eng.*, vol. 2025, Art. no. 5512389, 2025.
- [14] C. Liu, Y. Zhao, and J. Sun, “UAV target prediction and tracking based on KCF and Kalman filter,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2022, pp. 6712–6718.
- [15] M. Alshaer, T. Garouani, and S. Lecoeuche, “Vision-based UAV detection and tracking using deep learning and Kalman filter,” *Comput. Intell.*, vol. 41, no. 1, 2025.
- [16] R. Kumar and A. Singh, “Advanced algorithms for UAV tracking under start-stop and irregular motion,” *Sci. Rep.*, vol. 15, Art. no. 11342, 2025.
- [17] S. Lee and J. Park, “Deep recurrent reinforcement learning for intercept guidance under partial observability,” *Appl. Artif. Intell.*, vol. 38, no. 1, 2024.
- [18] H. Zhang, K. Wang, and L. Li, “UAV trajectory tracking via RNN-aided interacting multiple model Kalman filter,” arXiv:2312.15721, 2023.
- [19] B. Gaudet and R. Furfaro, “Reinforcement learning for angle-only intercept guidance of maneuvering targets,” arXiv:1906.02113, 2019.
- [20] D. A. Tarigan, M. Zarlis, and R. W. Sembiring, “Reverse tracking graph based on dynamic path planning,” *InfoTekJar: J. Nas. Inform. dan Teknol. Jaringan*, vol. 6, no. 1, 2021, doi: 10.30743/infotekjar.v6i1.4355.
- [21] P. Zarchan, *Tactical and Strategic Missile Guidance*, 6th ed. Reston, VA, USA: AIAA, 2012.
- [22] M. Guelman, “A qualitative study of proportional navigation,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-7, no. 4, pp. 637–643, 1971.
- [23] N. A. Shneydor, *Missile Guidance and Pursuit: Kinematics, Dynamics and Control*. Chichester, U.K.: Horwood, 1998.
- [24] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, 1968.
- [25] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numer. Math.*, vol. 1, pp. 269–271, 1959.
- [26] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.
- [27] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.
- [28] V. Mnih et al., “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, pp. 529–533, 2015.
- [29] T. P. Lillicrap et al., “Continuous control with deep reinforcement learning,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016.
- [30] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Pr