



Seminar Nasional LPPM UNSOED 2025

"Pengembangan Sumberdaya Perdesaan dan Kearifan Lokal Berkelanjutan"

8 Oktober 2025

Purwokerto

STUDI PENERAPAN ADVANCED DRIVER-ASSISTANT SYSTEMS (ADASs) TEKNOLOGI ADAPTIVE CRUISE CONTROL MENGUNAKAN MODEL PREDICTIVE CONTROL

**Arief Wisnu Wardhana^{1*}, Iwan Setiawan¹, Muhammad Syaiful Aliim¹, Mulki Indana Zulfa¹,
Agung Mubyarto¹, dan Hesti Susilawati¹**

¹Program Studi Teknik Elektro, Fakultas Teknik, Universitas Jenderal Soedirman

*Corresponding Author: arief.wardhana@unsoed.ac.id

ABSTRAK

Advanced Driver Assistance Systems (ADASs) merujuk pada teknologi yang mengotomatiskan, memfasilitasi, dan meningkatkan sistem dalam kendaraan untuk membantu pengemudi agar dapat berkendara dengan lebih baik dan aman. Salah satu jenis teknologi ADASs adalah teknologi Adaptive Cruise Control (ACC). Pada penelitian ini dilakukan sebuah studi untuk menerapkan teknologi ACC. Studi dilakukan dengan menggunakan alat bantu MATLAB–MathWorks Simulink yang merupakan alat bantu pemodelan yang mampu menganalisa sistem sistem yang bersifat dinamis. Untuk simulasi perangkat keras, digunakan Smart Robo Car 4WD dengan kontroler papan Arduino Uno yang diprogram menggunakan Arduino IDE. Hasil simulasi pada Simulink menunjukkan bahwa blok Adaptive Cruise Control berhasil mensimulasikan sistem ACC yang melacak kecepatan yang ditetapkan dan menjaga jarak aman dari kendaraan di depan dengan menyesuaikan akselerasi longitudinal kendaraan ego. Blok tersebut menghitung tindakan kontrol optimal sambil memenuhi batasan jarak, kecepatan, dan akselerasi yang aman menggunakan model predictive control (MPC). Sementara untuk simulasi perangkat keras, berhasil ditunjukkan bahwa Smart Rob Car 4WD berhasil menjalankan fungsi Adaptive Cruise Control (ACC) melalui program yang sudah ditanam. Beberapa data kuantitatif yang valid berupa data kecepatan kendaraan ego dan jarak antara kendaraan ego dan kendaraan di depannya berhasil diperoleh dan ditabelkan.

Kata kunci: *adaptive cruise control, MATLAB–Matworks, simulink, Arduino*

ABSTRACT

Advanced driver assistance systems (ADASs) refer to technologies that automate, facilitate, and improve systems in the vehicles in order to assist drivers for better and safer driving. One of the ADASs technologies is the Adaptive Cruise Control (ACC). This research conducted a study to implement the ACC technology. The study was conducted using MATLAB–MathWorks Simulink, a modeling tool capable of analyzing dynamic systems. For hardware simulation, a Smart Robo Car 4WD with an Arduino Uno board controller was used. It was programmed using the Arduino IDE. The simulation results in Simulink show that the Adaptive Cruise Control System block has successfully simulated an adaptive cruise control (ACC) system that tracks a set velocity and maintains a safe distance from a lead vehicle by



Seminar Nasional LPPM UNSOED 2025

"Pengembangan Sumberdaya Perdesaan dan Kearifan Lokal Berkelanjutan"

8 Oktober 2025

Purwokerto

adjusting the longitudinal acceleration of an ego vehicle. The block computes optimal control actions while satisfying safe distance, velocity, and acceleration constraints using model predictive control (MPC). Meanwhile, the hardware simulation showed that the Smart Robot Car 4WD has successfully run the Adaptive Cruise Control (ACC) function through the embedded program. Some valid quantitative data in the form of ego vehicle's speed and distance between the ego vehicle and the lead vehicle have been successfully obtained and tabulated.

Keywords: *adaptive cruise control, MATLAB-Matworks, simulink, Arduino*

PENDAHULUAN

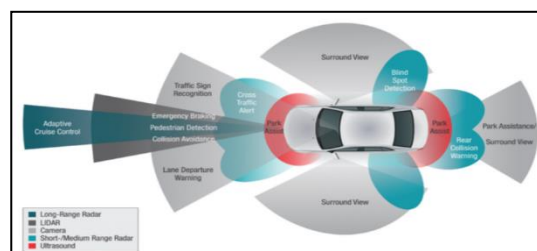
Keselamatan berkendara di jalan raya terutama bergantung pada faktor manusia. Survei literatur menunjukkan bahwa gangguan pengemudi merupakan penyebab utama semua kecelakaan lalu lintas. Selain itu, data menunjukkan bahwa insiden di jalan raya, termasuk tabrakan, terguling, dan kecelakaan lalu lintas lainnya, tetap menjadi penyebab utama kematian terkait pekerjaan (Holzmann, 2008).

Teknologi *Advanced Driver Assistance Systems* (ADASs) merujuk pada teknologi yang mengotomatiskan, memfasilitasi, dan meningkatkan sistem dalam kendaraan untuk membantu pengemudi agar bisa berkendara lebih baik dan aman (Texas Instruments, 2015).

Beberapa teknologi ADASs ditujukan untuk peningkatan keselamatan, dan beberapa lainnya ditujukan untuk fungsi kenyamanan. Fokus utama teknologi ADASs adalah berkontribusi pada faktor-faktor seperti manajemen keselamatan dan mengemudi otomatis bebas stres bagi pengemudi (Kathiresan dan Neelaveni, 2021).

Agar teknologi ADAS ini dapat berfungsi, penggunaan serangkaian sensor sangatlah penting. Sebagian besar fungsi ADASs tercanggih pada dasarnya merupakan sistem independen, dengan RADAR, LiDAR, kamera, dan ultrasound sebagai jenis sensor yang paling umum (Texas Instruments, 2015).

Bahkan, ke depannya terdapat tren yang semakin meningkat untuk menggunakan beberapa masukan sensor yang berbeda dan menggabungkannya guna membuat keputusan yang lebih akurat dan mengidentifikasi situasi kritis (Texas Instruments, 2015). Gambar 1 di bawah ini menunjukkan penggunaan berbagai jenis masukan sensor dan aplikasinya.





Seminar Nasional LPPM UNSOED 2025

"Pengembangan Sumberdaya Perdesaan dan Kearifan Lokal Berkelanjutan"

8 Oktober 2025

Purwokerto

Application/sensor type	Video	Infrared	Long range radar 76 to 81 MHz	Short/mid range radar 24 to 25 GHz/ 76 to 81 GHz	Ultrasound 40 kHz
Adaptive front light (AFL)	X				
Night vision (NV)	X	X			
Adaptive cruise control (ACC)	X	X	X		
Lane-departure warning (LDW)	X				
Low-speed ACC, emergency brake assist (EBA), lane-keep support (LKS)	X	X		X	
Pedestrian detection	X	X		X	
Blind-spot detection (BSD), rear collision warning (RCW), lane-change Assist (LCA)	X			X	
Park assist (PA)	X			X	X
Traffic-sign-recognition (TSR)	X				

Gambar 1. Sensor-sensor untuk ADASs

Pada penelitian ini dilakukan sebuah studi tentang penerapan berbagai sensor yang sesuai untuk teknologi ADASs. Penelitian ini membahas salah satu jenis teknologi ADASs yaitu Adaptive Cruise Control yang diimplementasikan menggunakan Model Predictive Control.

Terdapat dua kegiatan utama pada penelitian ini. Kegiatan pertama adalah melakukan simulasi ACC menggunakan Simulink – Simulasi dan Desain Berbasis Model – MATLAB – Mathworks. Kegiatan kedua adalah simulasi perangkat keras menggunakan mobil robot pintar 4WD yang telah diprogram sebelumnya untuk mendemonstrasikan kemampuan ACC ini. Pemrograman mobil robot pintar tersebut menggunakan Arduino IDE.

Terdapat banyak penelitian dan riset yang membahas tentang sistem bantuan keselamatan berkendara di jalan raya. Beberapa penelitian tentang sistem bantu keselamatan berkendara di jalan raya yang paling mendekati dengan penelitian ini dapat disebutkan sebagai berikut. Pertama, terdapat sebuah penelitian yang berjudul “Developing Adaptive Cruise Control Based on Fuzzy Logic Using Hardware Simulation” yang dikerjakan oleh Noor Cholis Basjaruddin, Kuspriyanto, Didin Saefudin, dan Ilham Khrisna Nugraha menunjukkan pengembangan pengontrol logika fuzzy untuk Adaptive Cruise Control. Metode inferensi fuzzy yang digunakan dalam penelitian ini adalah Mamdani. Hasil simulasi perangkat keras menggunakan mobil kendali jarak jauh menunjukkan bahwa pengontrol logika fuzzy dapat bekerja sesuai rancangan (Basjaruddin *et al.*, 2014).

Selanjutnya, terdapat penelitian lain berjudul ”Rancang Bangun Sistem Adaptive Cruise Control (ACC) dengan Metode Fuzzy Logic pada Prototype Robo Car” oleh Aulia Istiqomah, Luthfansyah Mohammad dan Muhammad Irham Agil Kurniawan membahas dan mengembangkan sistem ACC berbasis *Fuzzy Logic Controller* (FLC) menggunakan ESP8266 sebagai mikrokontroler utama, dengan tujuan mengoptimalkan pengendalian kecepatan kendaraan secara adaptif berdasarkan jarak terhadap objek di depannya (Istiqomah *et al.*, 2025). Kemudian terdapat juga penelitian berjudul “Adaptive Cruise Control of the Autonomous Vehicle Based on Sliding Mode Controller Using Arduino and Ultrasonic Sensor” yang dikerjakan oleh Rachid Alika, El Mehdi Mellouli, dan El Houssaine Tissir memfokuskan pada Adaptive Cruise Control pada kendaraan otonom (Alika *et al.*, 2024).

Kendaraan Otonom (*Autonomous Vehicle*) dan Tingkat Otomatisasi

Sebelum membahas lebih lanjut tentang berbagai sistem pendukung keselamatan berkendara di jalan raya, akan dibahas sedikit terlebih dahulu tentang istilah kendaraan otonom dan



Seminar Nasional LPPM UNSOED 2025

"Pengembangan Sumberdaya Perdesaan dan Kearifan Lokal Berkelanjutan"

8 Oktober 2025

Purwokerto

tingkat otomatisasi.

Kendaraan otonom (*Autonomous Vehicle*) adalah kendaraan yang dapat mengemudi sendiri tanpa input dari pengemudi. Sistem klasifikasi yang menjelaskan tingkat otomatisasi AV, mulai dari otomatis penuh hingga manual penuh, telah diterbitkan pada tahun 2014 oleh Society of Automotive Engineers (SAE) International, dan diadopsi oleh Departemen Perhubungan AS. Seperti yang diilustrasikan pada Gambar 2, Level 0 hingga Level 2 mengharuskan pengemudi manusia untuk selalu memantau kondisi berkendara (Alika *et al.*, 2024).

Level 0 berarti sedikit atau tanpa bantuan; pengemudi sepenuhnya mengendalikan mobil, sementara Level 1 menyediakan bantuan sederhana, seperti kendali kecepatan; pengemudi manusia dapat "lepas kaki" dan mulai mengalihkan sebagian tanggung jawab mengemudi ke mesin. Pada level 2 atau tahap otomatisasi parsial, kendali lateral dan longitudinal dapat diberikan oleh mesin dalam keadaan tertentu. Pengemudi dapat "lepas tangan" tetapi diharuskan untuk terus mengawasi lalu lintas dan siap untuk segera mengambil alih kendali kendaraan (Shi dan Li, 2025).

Level 0	Level 1	Level 2	Level 3	Level 4	Level 5
No system	"Feet off"	"Hands-off"	"Eyes-off"	"Brain-off"	No driver
No Assistance	Simple Assistance	Partial Automation	Conditional Automation	High Automation	Full Automation
					
Human	Transfer of responsibility				Machine

Gambar 2. Tingkat otomatisasi kendaraan

Transisi yang paling penting adalah antara otomatisasi parsial (Level 2) dan otomatisasi kondisional (Level 3). Kendaraan mengambil alih kendali lateral dan longitudinal dalam banyak situasi di level 3, sementara pengemudi manusia bisa "Eyes-off". Namun, pengemudi harus siap untuk mengambil alih kendali kendaraan sebagai cadangan, kapan pun diberitahu oleh mesin tepat waktu. Selanjutnya, seperti yang ditampilkan dalam Gambar 2 tersebut, perbedaan utama antara Level 4 (otomatisasi tinggi) dan Level 5 (otomatisasi penuh) adalah kemampuan sistem untuk menangani mode mengemudi terbatas tertentu versus semua mode mengemudi. Pada level 4, pengemudi manusia bisa "Brain-off", tetapi masih perlu mengambil tanggung jawab mengemudi pada mode mengemudi tertentu yang ditentukan. Pada Level 5 atau tahap otomatisasi penuh, mesin mengambil alih penuh mengemudi; tidak diperlukan pengemudi manusia (Shi dan Li, 2025).

Berbagai Sistem Teknologi *Advanced Driver-Assistance Systems*

Selanjutnya, Gambar 3 berikut mengilustrasikan empat elemen utama arsitektur *Advanced Driver-Assistance Systems* (ADASs) yang eksis saat ini: (1) kontrol longitudinal, (2) kontrol lateral, (3) sistem pemantauan kewaspadaan berkendara, dan (4) sistem alat bantu parkir.

Kontrol longitudinal seperti yang ditunjukkan di Gambar 3a sudah ada sejak diperkenalkan pertama kali pada tahun 1995, menggunakan embedded sensor di dalam mobil untuk



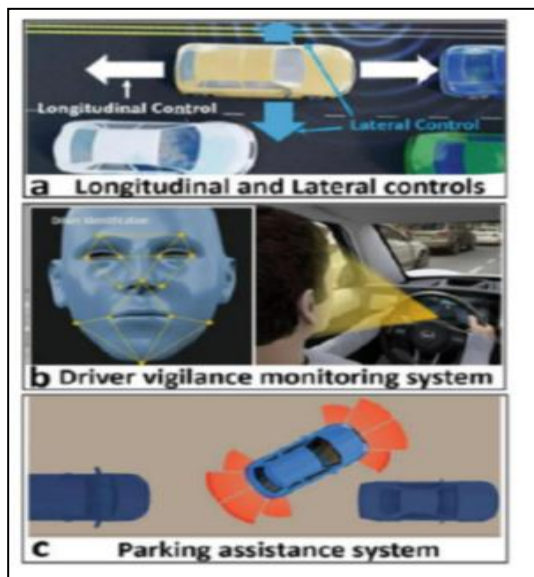
Seminar Nasional LPPM UNSOED 2025

"Pengembangan Sumberdaya Perdesaan dan Kearifan Lokal Berkelanjutan"

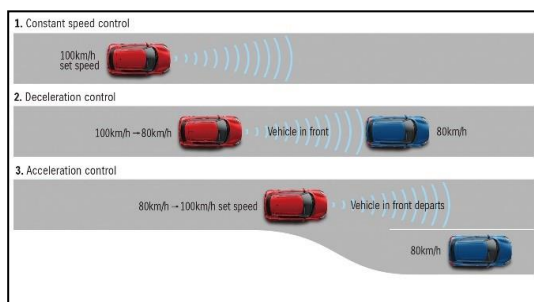
8 Oktober 2025

Purwokerto

mengukur jarak antara dua kendaraan di jalur yang sama. Kendaraan kelas menengah saat ini dilengkapi dengan *adaptive cruise control* dan sistem penghindaran tabrakan. Kedua sistem tersebut dapat memberi tahu pengemudi tentang risiko yang datang dari jalur yang sama, membantu mengendalikan kecepatan kendaraan, dan menawarkan tindakan kemudi yang mengelak. Lihat pada Gambar 4.



Gambar 3. Tingkat otomatisasi kendaraan



Gambar 4. Ilustrasi kontrol longitudinal pada ADASs

Kontrol lateral yang juga diilustrasikan pada Gambar 3a serupa dengan kontrol longitudinal di atas, yang dapat mendeteksi bahaya di lajur lateral, baik di sebelah kiri maupun di sebelah kanan kendaraan. Sistem ganti lajur dan sistem tetap lajur adalah dua sistem yang membantu tindakan ganti lajur terencana dan memperingatkan perubahan lajur yang tidak disengaja. Sistem ini dijalankan dengan mendeteksi kendaraan yang datang atau situasi berbahaya lainnya selama pergantian lajur terencana dan mengenali penyimpangan jalur kendaraan secara tiba-tiba dari lajur normal (Shi dan Li, 2025).

Distraksi saat berkendara merupakan salah satu penyebab utama kecelakaan, terutama bagi



Seminar Nasional LPPM UNSOED 2025

"Pengembangan Sumberdaya Perdesaan dan Kearifan Lokal Berkelanjutan"

8 Oktober 2025

Purwokerto

pengemudi muda. Seperti yang diilustrasikan pada Gambar 3b, sistem pemantauan kewaspadaan pengemudi dapat mengevaluasi tingkat stres, kelelahan, dan kemarahan pengemudi melalui kamera dan sensor yang tertanam di kokpit, memantau wajah dan ekspresi pengemudi secara langsung (real-time). Sistem ini dapat digunakan untuk mengungkap potensi masalah keselamatan akibat emosi pengemudi yang tidak sehat.

Sistem asisten parkir yang ditunjukkan pada Gambar 3c, saat ini merupakan teknologi ADASs terancang, karena dapat menyediakan parkir otomatis tanpa perlu kemudi atau kehadiran pengemudi. Kamera di depan dan belakang mobil, bersama dengan sensor lateral dan longitudinal lainnya, menyediakan lapisan persepsi atau input untuk mobil. Sistem cerdas yang menggunakan algoritma pembelajaran mesin menganalisis data input, memberikan keputusan yang tepat, dan secara otomatis memarkir mobil di lokasi yang diinginkan (Shi dan Li, 2025).

Adaptive Cruise Control (ACC)

Adaptive cruise control menggunakan sebuah kamera yang melihat ke depan atau kombinasi kamera yang melihat ke depan dan teknologi radar (tergantung jenis kendaraannya) untuk memantau kendaraan yang diikuti. Fitur ini akan membantu mengikuti kendaraan di depan pada jarak yang bisa pilih (jauh, sedang, atau dekat) saat kita mengemudi. Ini membantu mengurangi kebutuhan untuk sering mengerem maupun berakselerasi.

Jika sistem tidak mendeteksi kendaraan di depan, *adaptive cruise control* akan bekerja seperti cruise control biasa dan mempertahankan kecepatan jelajah yang kita pilih. Ketika sistem mendeteksi kendaraan di depan kita di jalur kita, sistem akan secara otomatis memperlambat atau mempercepat kendaraan kita untuk mempertahankan jarak yang kita pilih.

Adaptive cruise control tidak menyediakan otonomi penuh: sistem ini hanya menyediakan sedikit bantuan kepada pengemudi, tetapi tidak akan mengemudikan mobil sendiri.

Komponen Teknologi Sistem ACC

Sensor radar biasanya merupakan inti dari sistem *adaptive cruise control*. Dipasang di bagian depan kendaraan, sistem ini secara permanen memantau jalan di depan. Selama jalan di depan kosong (bersih), ACC mempertahankan kecepatan yang diatur oleh pengemudi. Jika sistem mendeteksi kendaraan yang lebih lambat dalam jangkauan deteksinya, sistem akan mengurangi kecepatan secara perlahan dengan melepaskan pedal gas atau mengaktifkan sistem kontrol rem secara aktif. Jika kendaraan di depan menambah kecepatan atau berpindah jalur, ACC secara otomatis akan mempercepat laju kendaraan ke kecepatan yang diinginkan pengemudi.

Untuk meningkatkan kenyamanan dan keamanan fungsi ini, kamera multifungsi dapat dipasang bersama dengan sensor radar. Dengan demikian, misalnya, berkat akurasi pengukuran lateral kamera multifungsi, ACC dapat mendeteksi kendaraan yang memasuki jalur pengemudi – baik yang direncanakan maupun tidak – jauh lebih awal, sehingga memungkinkan sistem untuk merespons secara lebih dinamis. Untuk pemahaman situasi yang lebih baik dan lebih akurat, data dari sensor radar dan kamera dapat digabungkan.

METODE PENELITIAN



Seminar Nasional LPPM UNSOED 2025

"Pengembangan Sumberdaya Perdesaan dan Kearifan Lokal Berkelanjutan"

8 Oktober 2025

Purwokerto

Waktu dan Tempat Penelitian

Penelitian dilaksanakan di tempat tinggal penulis dengan menggunakan satu perangkat Laptop Acer Aspire 5 dengan prosesor Intel CORE i5. Pada Laptop tersebut sudah ter install software MATLAB MathWorks yang akan digunakan untuk simulasi Simulink. Juga sudah ter install software Arduino IDE 2.3.6 yang akan digunakan untuk memprogram smart robo car.

Penelitian dilakukan mulai bulan April 2025 sampai dengan bulan Agustus 2025. Penelitian terdiri dari 2 tahap. Tahap pertama adalah simulasi Adaptive Cruise Control menggunakan MATLAB MathWorks – Simulink. Tahap kedua adalah simulasi hardware menggunakan Smart Robo Car 4WD.

Sistem Adaptive Cruise Control Menggunakan Model Predictive Control (MPC) Menggunakan Pemodelan Matlab – Matworks Simulink

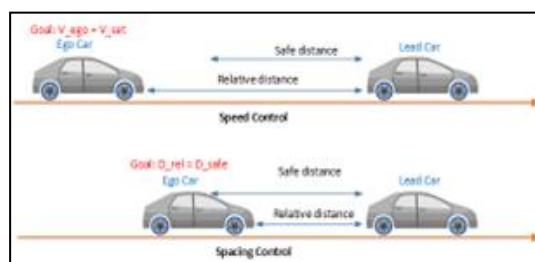
Penelitian ini berkaitan dengan sistem yang sifatnya dinamis, melibatkan model kendaraan yang akan menyesuaikan pergerakan mereka sesuai dengan keadaan lingkungan di sekitarnya.

MATLAB – MathWorks Simulink adalah lingkungan pemrograman grafis berbasis MATLAB untuk pemodelan, simulasi, dan analisis sistem dinamis multidomain. Antarmuka utamanya adalah alat diagram blok grafis dan serangkaian pustaka blok yang dapat disesuaikan. Simulink menawarkan integrasi yang erat dengan lingkungan MATLAB lainnya dan dapat dijalankan di MATLAB atau ditulis darinya.

Penjelasan Blok ACC

Blok *Adaptive Cruise Control System* pada MATLAB mensimulasikan sebuah sistem *Adaptive Cruise Control (ACC)* yang melacak kecepatan yang sudah ditentukan dan mempertahankan jarak aman dari *lead car* dengan menyesuaikan akselerasi longitudinal *ego car*. Blok ini menghitung aksi kendali optimal sambil memenuhi jarak aman, kecepatan aman, dan juga keterbatasan besarnya akselerasi, dengan menggunakan *model predictive control (MPC)* (lihat Gambar 5) (The MathWorks, Inc., 2025).

Sebuah kendaraan (dinamakan *ego car*) yang dilengkapi dengan *Adaptive Cruise Control (ACC)* memiliki sebuah sensor, misalnya radar, yang akan mengukur jarak ke kendaraan di depannya (dinamakan *lead car*), $D_{relative}$, di lintasan yang sama. Sensor juga akan mengukur kecepatan relatif dari *lead car*, $V_{relative}$.



Gambar 5. Ilustrasi sistem *Adaptive Cruise Control*



Seminar Nasional LPPM UNSOED 2025

"Pengembangan Sumberdaya Perdesaan dan Kearifan Lokal Berkelanjutan"

8 Oktober 2025

Purwokerto

Sistem ini beroperasi dengan dua mode operasi berikut ini (The MathWorks, Inc., 2025): Mode pertama adalah mode kontrol kecepatan dimana *ego car* akan bergerak dengan kecepatan yang ditentukan oleh pengemudi. Mode kedua adalah mode kontrol jarak dimana *ego car* akan menjaga jarak aman dari *lead car*.

Sistem ACC akan menentukan mode mana dari kedua mode di atas yang akan digunakan, berbasis pada hasil pengukuran real-time radar. Misalnya, jika *lead car* terlalu dekat, sistem ACC akan beralih dari mode kontrol kecepatan ke mode kontrol jarak. Demikian pula, jika *lead car* menjadi lebih jauh, maka sistem ACC akan beralih dari mode kontrol jarak ke mode kontrol kecepatan. Dengan kata lain, sistem ACC membuat mobil *ego car* melaju dengan kecepatan yang ditentukan pengemudi selama keduanya berada dalam jarak aman (The MathWorks, Inc., 2025).

Jika $D_{relative} \geq D_{safe}$, maka mode kontrol kecepatan yang aktif. Tujuan pengendalian nya adalah untuk melacak kecepatan yang ditetapkan pengemudi, V_{set} (Lihat bagian atas dari Gambar 5). Jika $D_{relative} < D_{safe}$, maka mode kontrol jarak yang aktif. Tujuan pengendaliannya adalah untuk menjaga jarak aman, D_{safe} (Lihat bagian bawah dari Gambar 5).

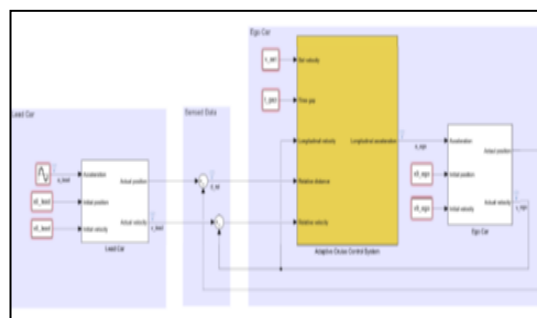
Penjelasan Model Predictive Control (MPC)

Kendali prediktif model (MPC) merupakan teknik kendali optimal di mana tindakan kendali yang dihitung meminimalkan fungsi biaya untuk sistem dinamis yang dibatasi pada cakrawala yang terbatas dan surut.

Pada setiap langkah waktu, pengontrol MPC menerima atau memperkirakan kondisi terkini pabrik. Kemudian, pengontrol menghitung urutan tindakan kontrol yang meminimalkan biaya di masa mendatang dengan memecahkan masalah optimasi terbatas yang bergantung pada model pabrik internal dan bergantung pada kondisi sistem saat ini. Pengontrol kemudian hanya menerapkan tindakan kontrol pertama yang telah dihitung ke pabrik, mengabaikan tindakan kontrol berikutnya. Pada langkah waktu berikutnya, proses ini berulang (The MathWorks, Inc., 2025).

Simulink Model di MATLAB untuk Lead Car dan Ego Car

Dinamika untuk *lead car* dan *ego car* pada Simulink dimodelkan seperti yang ditunjukkan di Gambar 6 berikut (The MathWorks, Inc., 2025).



Gambar 6. Model simulink untuk Lead Car dan Ego Car



Seminar Nasional LPPM UNSOED 2025

"Pengembangan Sumberdaya Perdesaan dan Kearifan Lokal Berkelanjutan"

8 Oktober 2025

Purwokerto

Posisi aktual (posisi absolut) dari *lead car* dan posisi aktual dari *ego car* akan di sum, dan akan menghasilkan informasi $D_{relative}$. Merupakan jarak dalam hitungan meter antara *lead vehicle* dan *ego vehicle*. Berupa skalar positif. Untuk menghitung sinyal ini, kurangi posisi *ego car* dari posisi *lead car*.

Kecepatan aktual (kecepatan absolut) dari *lead car* dan kecepatan aktual dari *ego car* akan di sum dan akan menghasilkan informasi $V_{relative}$ (perbedaan kecepatan antara *lead vehicle* dan *ego vehicle*) dalam hitungan meter. Berupa skalar. Untuk menghitung sinyal ini, kurangi kecepatan *ego car* dari kecepatan *lead car*. Selain itu, juga akan menghasilkan $V_{ego-longitudinal}$ (kecepatan longitudinal *ego vehicle*).

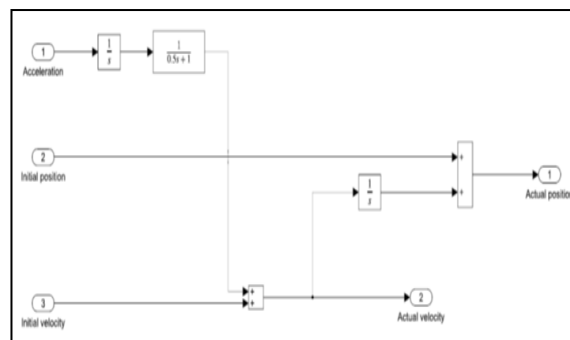
Untuk mendekati suasana berkendara yang sebenarnya, percepatan dari *lead car* dibuat bervariasi sesuai dengan gelombang sinus selama simulasi. Blok sistem *Adaptive Cruise Control* nantinya akan menghasilkan sebuah sinyal kendali percepatan untuk *ego car* (The MathWorks, Inc., 2025).

Tentukan waktu sampel, T_s , dan durasi simulasi, T , dalam hitungan detik.

$$T_s = 0.1;$$

$$T = 140;$$

Untuk *ego car* maupun *lead car*, dinamika antara akselerasi dan kecepatan dimodelkan sebagai $G = \frac{1}{s(0.5s+1)}$ yang mendekati dinamika *throttle body* dan inersia kendaraan seperti terlihat di Gambar 7.



Gambar 7. Dinamik model antara percepatan dan kecepatan

Integral dari percepatan akan menghasilkan kecepatan. Dan integral dari kecepatan akan menghasilkan posisi. Ini berlaku baik untuk *ego car* maupun *lead car*.

Selanjutnya, tentukan model linier untuk *ego car*.

$$G_{ego} = tf(1,[0.5,1,0]);$$

Kemudian tentukan posisi awal dan kecepatan awal untuk *lead car* dan untuk *ego car* sebagai berikut

$$\begin{aligned} x_{0_lead} &= 100; \quad \% \text{ initial position for lead car (m)} \\ v_{0_lead} &= 10; \quad \% \text{ initial velocity for lead car (m/s)} \end{aligned}$$



Seminar Nasional LPPM UNSOED 2025

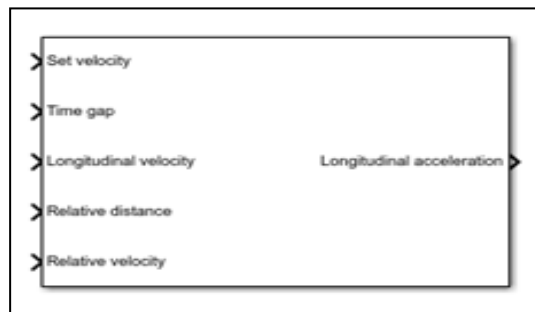
"Pengembangan Sumberdaya Perdesaan dan Kearifan Lokal Berkelanjutan"

8 Oktober 2025

Purwokerto

```
 $x_{0\_ego} = 10;$  % initial position for ego car (m)  
 $v_{0\_ego} = 5;$  % initial velocity for ego car (m/s)
```

Sistem ACC dimodelkan menggunakan blok *Adaptive Cruise Control System Block* di Simulink sebagai Gambar 8 berikut.



Gambar 8. Model blok sistem *Adaptive Cruise Control* di simulink

Terlihat pada Gambar 8 di atas, beberapa input untuk blok sistem ACC adalah sebagai berikut (The MathWorks, Inc., 2025):

- Kecepatan *ego vehicle* yang ditentukan oleh pengemudi V_{set}
- Gap waktu T_{gap} (Safe time gap dalam hitungan detik antara *lead vehicle* dan *ego vehicle*).
 T_{gap} ini digunakan untuk menghitung batasan jarak aman minimum.
- Kecepatan longitudinal *ego car* V_{ego}
- Jarak relatif ke *lead car* $D_{relative}$ (dari radar).
- Kecepatan relative terhadap *lead car* $V_{relative}$ (dari radar)

Output dari sistem ACC adalah berupa percepatan longitudinal *ego car*. Jarak aman antara *lead car* dan *ego car* adalah merupakan fungsi dari kecepatan *ego car*, V_{ego} . Secara default, *model predictive controller* menghitung batasan jarak aman; yaitu, jarak relatif minimum antara *lead vehicle* dan *ego vehicle*, dengan persamaan berikut ini:

$$D_{safe} = D_{default} + T_{gap} \times V_{ego}$$

dimana $D_{default}$ adalah jarak default berhenti. Sinyal input T_{gap} adalah safe time gap dalam hitungan detik antara *lead vehicle* dan *ego vehicle*. Sinyal input T_{gap} ini digunakan untuk menghitung batasan jarak aman minimum (The MathWorks, Inc., 2025). Kemudian, V_{ego} adalah sinyal input kecepatan longitudinal dari *ego car*.

Tentukan nilai untuk $D_{default}$, dalam hitungan meter, dan T_{gap} , dalam hitungan detik



Seminar Nasional LPPM UNSOED 2025

"Pengembangan Sumberdaya Perdesaan dan Kearifan Lokal Berkelanjutan"

8 Oktober 2025

Purwokerto

sebagai berikut:

$$t_{gap} = 2.6;$$

$$D_{default} = 10;$$

Kemudian, kecepatan yang di set oleh pengemudi ditentukan, dalam satuan m/s .

$$v_{set} = 35 \text{ m/s}$$

Kecepatan sebesar $v_{set} = 35 \text{ m/s}$ adalah setara dengan kecepatan sebesar $v_{set} = 126 \text{ km/h}$. Jadi, dengan menggunakan nilai nilai $D_{default} = 10 \text{ m}$, $T_{gap} = 2.6 \text{ detik}$, dan $V_{ego} = 35 \text{ m/s}$ seperti di atas, dapat dihitung bahwa

$$D_{safe} = 10 + (2.6 \times 35) = 101 \text{ m}$$

Selama $D_{relative} \geq D_{safe}$, maka *ego car* akan menggunakan kecepatan yang sudah ditetapkan oleh pengemudi $V_{set} = 35 \text{ m/s}$. Sebaliknya apabila $D_{relative} \leq D_{safe}$, maka *ego car* akan menjaga jarak aman dari *lead car* dengan menyesuaikan kecepatannya.

Mengingat keterbatasan fisik dinamika kendaraan, akselerasi dibatasi pada jangkauan $[-3, 2] \left(\frac{m}{s^2}\right)$.

$$amin_ego = -3;$$

$$amax_ego = 2;$$

Untuk contoh ini, parameter default blok Sistem *Adaptive Cruise Control* sesuai dengan parameter simulasi. Jika parameter simulasi berbeda dari nilai default, maka parameter blok perlu diperbaharui sebagaimana mestinya (The MathWorks, Inc., 2025). Misalnya apabila ternyata ditetapkan bahwa salah satu parameter yaitu $T_{gap} = 5$ (bukan nilai default 2.6). Dimana pada situasi cuaca buruk atau situasi kecepatan kecepatan pada jalan bebas hambatan, maka T_{gap} harus dinaikan menjadi 4 – 6 detik untuk memperhitungkan faktor jarak pandang dan jarak berhenti yang lebih panjang.

HASIL DAN PEMBAHASAN

Simulasi MATLAB – MathWorks Simulink

Untuk melakukan simulasi, kita harus menjalankan *Run the simulation* pada command window dengan command berikut ini

$$\text{sim}(\text{mdl})$$

Untuk melihat hasil plot, kita akan menjalankan *Plot the simulation result* dengan menggunakan command berikut ini

$$\text{mpcACCplot}(\text{logout}, D_{\text{default}}, t_{\text{gap}}, v_{\text{set}})$$



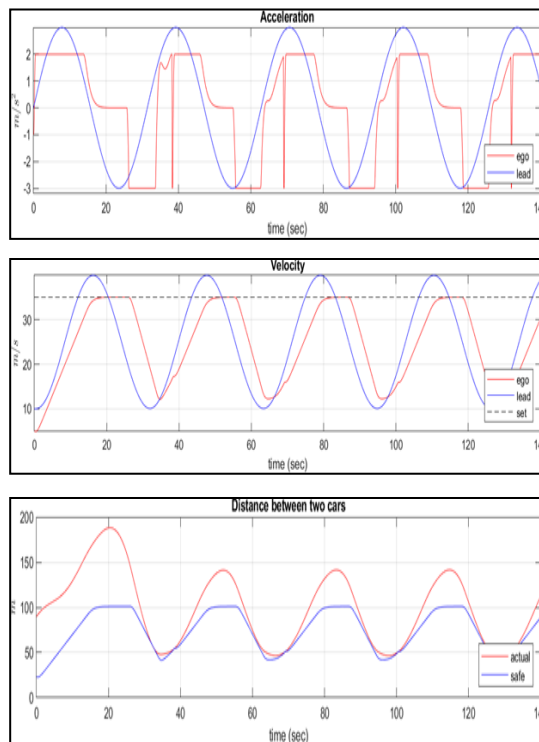
Seminar Nasional LPPM UNSOED 2025

"Pengembangan Sumberdaya Perdesaan dan Kearifan Lokal Berkelanjutan"

8 Oktober 2025

Purwokerto

Pada MATLAB, akan dihasilkan simulasi seperti yang ditunjukkan pada Gambar 9 berikut ini.



Gambar 9. Plot hasil simulasi

Analisa Simulasi MATLAB – MathWorks Simulink

Durasi simulasi keseluruhan adalah selama 140 detik. Pada 3 detik pertama (detik 0 sampai detik 3), untuk mencapai kecepatan yang ditetapkan pengemudi, the ego car akan berakselerasi dengan kecepatan penuh (The MathWorks, Inc., 2025).

Untuk selanjutnya, ketiga plot di atas dapat dianalisa sebagai berikut. Dari mulai detik ke 3 sampai ke 13, the lead car mempercepat secara perlahan lahan. Sebagai akibatnya, untuk menjaga jarak aman dengan the lead car, maka the ego car berakselerasi dengan kecepatan lebih lambat

Selanjutnya, mulai detik ke 13 hingga detik ke 25 detik, the ego car mempertahankan kecepatan yang ditentukan pengemudi, seperti yang ditunjukkan pada plot kecepatan. Namun, ketika the lead car mengurangi kecepatan, kesalahan jarak mulai mendekati 0 setelah 20 detik.



Seminar Nasional LPPM UNSOED 2025

"Pengembangan Sumberdaya Perdesaan dan Kearifan Lokal Berkelanjutan"

8 Oktober 2025

Purwokerto

Kemudian, dari detik ke 25 hingga detik ke 45, *the lead car* melambat lalu berakselerasi lagi. *The ego car* berusaha menjaga jarak aman dari *the lead car* dengan menyesuaikan kecepatannya, seperti yang ditunjukkan pada plot jarak.

Dari detik ke 45 sampai detik ke 56, spacing error nya ada jauh di atas 0. Oleh karenanya, *the ego car* mencapai kecepatan yang ditetapkan pengemudi lagi.

Terakhir, mulai dari detik ke 56 sampai detik ke 76, adalah mengulang kembali sekuensi perlambatan / percepatan seperti yang terjadi pada interval detik ke 25 sampai detik ke 45 (The MathWorks, Inc., 2025).

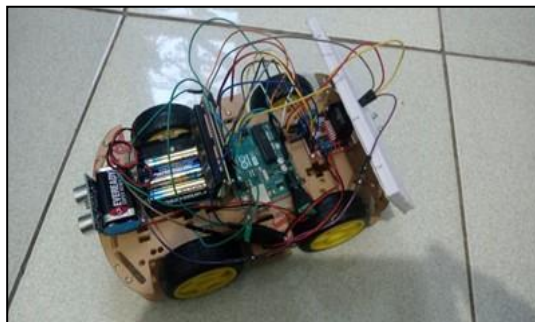
Analisa selanjutnya yang bisa dibuat adalah sebagai berikut. Sepanjang simulasi (bisa dilihat mulai dari detik 0 sampai detik 140 pada plot jarak), pengontrol akan memastikan bahwa jarak aktual antara kedua kendaraan lebih besar daripada jarak aman yang ditetapkan.

Ketika jarak aktual cukup besar, jauh di atas jarak aman, pengontrol memastikan bahwa *the ego car* mengikuti kecepatan yang ditetapkan pengemudi (sebesar 35 m/s). Bisa dilihat pada plot kecepatan dan plot jarak

Bisa dilihat juga di sepanjang simulasi, bahwa jarak aman antara *lead car* dan *ego car*, D_{safe} , adalah merupakan fungsi dari kecepatan *ego car*. Bisa dilihat pada plot kecepatan dan plot jarak.

Simulasi Hardware

Dilakukan menggunakan Smart Rob Car 4WD seperti yang ditunjukkan di Gambar 10 di bawah ini



Gambar 10. Smart Robo Car 4WD

Implementasi akan terdiri dari dua bagian berkaitan dengan mode operasi dari sistem *Adaptive Cruise Control* nya.

Implementasi yang pertama adalah berkaitan dengan keadaan pada saat $D_{relative} \geq D_{safe}$, sehingga yang diaktifkan adalah sistem kendali kecepatan. Di sini, tujuan kontrol adalah untuk melacak kecepatan yang sudah ditetapkan oleh pengemudi, V_{set} .

Implementasi yang kedua adalah berkaitan dengan keadaan pada saat $D_{relative} < D_{safe}$, sehingga yang diaktifkan adalah sistem kendali jarak. Tujuan pengendaliannya adalah untuk menjaga jarak aman, D_{safe}



Seminar Nasional LPPM UNSOED 2025

"Pengembangan Sumberdaya Perdesaan dan Kearifan Lokal Berkelanjutan"

8 Oktober 2025

Purwokerto

Program Arduino untuk Kendali Kecepatan dan Kendali Jarak

Pemrograman untuk mode kendali kecepatan melibatkan pemrograman mikrokontroler Atmega328P pada papan Arduino UNO menggunakan Arduino IDE.

Bagian awal program terdiri dari memasukkan library untuk LCD I2C dan untuk motor driver L298NX2. Dilanjutkan dengan inisialisasi LCD I2C, DC motor driver, dan sensor ultrasonik. Bagian selanjutnya adalah program untuk kendali kecepatan dan program untuk kendali jarak.

Tabel 1 berikut ini menampilkan hasil dari simulasi hardware. Hasil tersebut dibandingkan dengan hasil pada Simulink.

Tabel 1. Hasil plot simulasi simulink dan implementasi pada Smart Robo Car 4WD

Jarak <i>lead car</i> dan <i>ego</i> <i>car</i> di Simulink (m)	Kecepatan <i>ego car</i> pada Simulink		Kecepatan di Program Arduino	Jarak antara <i>lead car</i> dan <i>ego car</i> pada Program Arduino(cm)
	(m/s)	(km/h)		
187.746	35.00	126.00	126	187
179.952	35.00	126.00	126	180
170.776	35.00	126.00	126	170
160.592	35.00	126.00	126	160
149.544	35.00	126.00	126	149
140.093	35.00	126.00	126	140
120.259	35.00	126.00	126	120
110.403	33.09	119.13	119	110



Seminar Nasional LPPM UNSOED 2025

"Pengembangan Sumberdaya Perdesaan dan Kearifan Lokal Berkelanjutan"

8 Oktober 2025

Purwokerto

89.803	26.53	95.50	96	90
79.949	24.43	87.94	88	80
70.780	22.02	79.26	79	70
60.640	18.42	66.30	66	60
50.135	12.75	45.90	46	50
-	-	-	40	45
-	-	-	30	35
-	-	-	20	25

KESIMPULAN

Simulasi sistem *Adaptive Cruise Control* menggunakan *Model Predictive Control* dengan bantuan MathWorks MATLAB Simulink sudah berhasil dilakukan dan menghasilkan beberapa plot yaitu plot percepatan, kecepatan, dan jarak antara *ego car* dan *lead car* yang bisa dianalisa. Dari plot plot tersebut dapat dianalisa reaksi dari *ego car* terhadap perilaku *lead car* yang ada di depannya. Prototipe simulasi hardware dari sistem *Adaptive Cruise Control* juga sudah berhasil dikembangkan dan sudah dilakukan diuji. Hasil pengujian menunjukkan bahwa prototipe simulasi hardware bisa berfungsi sesuai dengan program. Prototipe simulasi berhasil berfungsi sebagai vehicle yang mempunyai fitur *Adaptive Cruise Control*, tentu saja dengan keterbatasan yang dimiliki oleh komponen-komponennya. Untuk saran berikutnya bisa digunakan sensor ultrasonic atau sensor radar atau sensor LiDAR yang memiliki daya jangkauan yang lebih jauh dan lebih luas. Kemudian digunakan vehicle dengan ukuran yang mendekati ukuran sebenarnya. Saran selanjutnya adalah dengan melakukan penelitian masih pada bidang *Adaptive Cruise Control*. Yaitu untuk menerapkan *Adaptive Cruise Control* berbasis fusi sensor untuk kendaraan yang melaju di jalan melengkung menggunakan fusi sensor.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Lembaga Penelitian dan Pengabdian Masyarakat Universitas Jenderal Soedirman atas pembiayaan penelitian ini melalui Dana Penelitian BLU Unsoed 2025 skema penelitian Riset Peningkatan Kompetensi (RPK) dengan nomor kontrak 14.422/UN23.34/PT.01/V/2025



Seminar Nasional LPPM UNSOED 2025

"Pengembangan Sumberdaya Perdesaan dan Kearifan Lokal Berkelanjutan"

8 Oktober 2025

Purwokerto

DAFTAR PUSTAKA

- Alika, R., E.M. Mellouli, dan E.H. Tissir. 2024. Adaptive Cruise Control of the Autonomous Vehicle Based on Sliding Mode Controller Using Arduino and Ultrasonic Sensor. *Journal of Robotics and Control (JRC)* 5(1). <https://doi.org/10.18196/jrc.v5i1.20519>
- Basjaruddin, N.C., Kuspriyanto, D. Saefudin, dan I.K. Nugraha. 2014. Developing Adaptive Cruise Control Based on Fuzzy Logic Using Hardware Simulation. *International Journal of Electrical and Computer Engineering (IJECE)* 4(6): 944–951.
- Holzmann, F. 2008. *Adaptive Cooperation between Driver and Assistant System: Improving Road Safety*. Regensburg: Springer-Verlag, hlm. 3–4.
- Istiqomah, A., L. Mohammad, dan M.I.A. Kurniawan. 2025. Rancang Bangun Sistem Adaptive Cruise Control (ACC) dengan Metode Fuzzy Logic pada Prototype Robo Car. *Jurnal Otomasi Kontrol dan Instrumentasi* 17(1). <https://doi.org/10.5614/joki.2025.17.1.4>.
- Kathires, M., dan R. Neelaveni. 2021. *Automotive Embedded Systems – Key Technologies, Innovations, and Applications*. Springer Nature, hlm. 3–4.
- Shi, H., dan Y. Li. 2022. *Advanced Driver Assistant Systems and Autonomous Vehicles: From Fundamentals to Applications*. Springer-Verlag, hlm. 1–2, 17–20.
- Texas Instruments. 2015. *Advanced Driver Assistance (ADAS) Solutions Guide*. Report No. SLYY044A. Texas Instruments.
- The MathWorks, Inc. 2025. *Adaptive Cruise Control System*. MATLAB Help Center. <https://au.mathworks.com/help/mpc/ref/adaptivecruisecontrolsystem.html>.
- The MathWorks, Inc. 2025. *Adaptive Cruise Control Using Model Predictive Controller*. MATLAB Help Center. <https://au.mathworks.com/help/mpc/ug/adaptive-cruise-control-using-model-predictive-controller.html>.
- The MathWorks, Inc. 2025. *What Is Model Predictive Control?* MATLAB Help Center. <https://au.mathworks.com/help/mpc/gs/what-is-mpc.html>.