RESEARCH ARTICLE
Journal of Field Robotics

Hierarchical Rule-Base Reduction Fuzzy Control for Constant Velocity Path Tracking of a Differential Steer Vehicle

Maxwel T. Cichon[1] | Samuel R. Dekhterman[1] | William R. Norris PhD[1] | Dustin Nottage PhD[2] | Ahmet Soylemezoglu PhD[2]

1 | ACKNOWLEDGMENTS

The authors are grateful to: 1- The financial support by and approval of the U.S. Army Corps of Engineers Engineering Research and Development Center, Construction Engineering Research Laboratory under grant number W9132T-19-C-0004; 2- Orson Wang for his assistance in transporting the test platform to the test site.

[1]Department of Industrial and Enterprise Systems Engineering, University of Illinois Urbana-Champaign, IL, 61801, USA

[2]Construction Engineering Research Laboratory, U.S. Army Corps of Engineers Engineering Research and Development Center, IL, 61822, USA

Correspondence
Samuel R. Dekhterman, Department, Department of Industrial and Enterprise Systems Engineering, University of Illinois Urbana-Champaign, IL, 61801, USA
Email: srd2@illinois.edu

Present address
[†]Department of Industrial and Enterprise Systems Engineering, University of Illinois Urbana-Champaign, IL, 61801, USA

Funding information
U.S. Army Corps of Engineers Engineering Research and Development Center, Construction Engineering Research Laboratory, Grant: W9132T-19-C-0004

A new form of waypoint navigation controller for a skid-steer vehicle is presented, which consisted of a multiple input-single output nonlinear fuzzy angular velocity controller. The membership functions of the fuzzy controller employed a trapezoidal structure with a completely symmetric rule-base. Notably, Hierarchical Rule-Base Reduction (HRBR) was incorporated into the controller to select only the rules most influential on state errors. This was done by selecting inputs/outputs and generating a hierarchy of inputs using a Fuzzy Relations Control Strategy (FRCS). Similar to some traditional fuzzy controllers, the system provided coverage for the global operating environment. However, a rule for every possible combination of variables and states was no longer necessary. Consequently, HRBR fuzzy controllers effectively increase both the number of inputs and their associated fidelity without the rule-base dramatically increasing. To contextualize the performance of the

controller, a background on vehicle dynamic modeling methodologies and an in-depth explanation of the related simulation model are provided. An examination of the proposed controller is then completed employing test courses. The test courses examine the effects of steering disturbance, phase lag, and overshoot as expressed in Root Mean Square Error (RMSE), Max Error (ME), and Course Completion Time (CCT). Finally, simulation and experimental results for the controller's performance were compared with a state-of-the-art waypoint navigation vehicle controller, geometric pure pursuit. The fuzzy was found to outperform the pure pursuit experimentally by 52.1 percent in RMSE, 26.8 percent in ME, and 1.07 percent in CCT, on average, validating the viability of the controller.

KEYWORDS
Differential steer vehicle, fuzzy logic, pure pursuit, ROS, waypoint navigation

2 | INTRODUCTION

Industries such as construction, agriculture, mining, etc. are increasingly relying on autonomous vehicles to traverse off-road environments. As such, control effects have been employed to help these vehicles operate as intended. In this paper, the control efforts for skid steer autonomous vehicles will be examined.

A skid steer vehicle, when represented as a unicycle model (Hellström, 2011), has no A matrix and the B matrix directly maps the vehicle dynamics to the control action. This structure is directly at odds with Model-based approaches, like LQR, MPC, and H-infinity. These controllers use the vehicle's A matrix and control error signal, to develop an optimal and/or robust target trajectory and a corresponding control action. (Vinodh Kumar & Jerome, 2013)(Tashiro, 2013) (Khelfi & Abdessameud, 2007). Several other control approaches include:

Frequency domain controllers, like P, PI, PD, or PID control, being inherently SISO, can not account for multiple objectives at the same time and require a pre-processing, mid-level path planning component to select waypoints (Majid, Mohamed, & Basri, 2016). The stability dynamics associated with this approach will also vary heavily with travel time, speed, and terrain (Campbell, 2007). Stability and performance criteria can still be guaranteed, but this requires a sensitivity analysis and rigorous system identification, only to be valid on hard, flat ground (Pentzer, Brennan, & Reichard, 2014).

Sliding mode controllers operate in a binary fashion by prescribing either a maximal or minimal control effort to drive the desired error state to a sliding manifold with a zero-error state (Young, Utkin, & Ozguner, 1999). Sliding mode controllers have been demonstrated to work to an extent on skid steer autonomous land vehicles (Jong-Min Yang & Jong-Hwan Kim, 1999), but the steady-state oscillations have kept them from widespread use.

Learning-based controllers, (Kuutti, Bowden, Jin, Barber, & Fallah, 2019), that create a multi-variable, nonlinear,

sensor input-control output mapping have become popular for autonomous vehicle control applications (Ferrari & Stengel, 2005). However, vehicles can become unstable (Yi, Song, Zhang, & Goodwin, 2007) when presented with disturbances outside of their training space, such as unexpected changes to the vehicle dynamics, ground contact physics, or unforeseen sensor measurements. As such, off-road environments pose a challenge.

Geometric controllers determine the optimal control action based on the manifold defined by the geometric based constraints on the vehicle and its error state (Sachkov, 2019). For differential and skid steer vehicles, the most common form of geometric control is pure pursuit (Samuel, Hussein, & Mohamad, 2016b). These controllers are robust, so long as the target point is far enough away from the vehicle to account for the maximum system time delays and any discrepancies between the real-world vehicle dynamics and model dynamics (Murphy, 1994). As such, pure pursuit was the baseline controller used in this paper.

Given the issues presented, fuzzy control was investigated. Historical background on the use of fuzzy logic in general and applications in autonomous vehicle control can be found in (Bělohlávek, Dauben, & Klir, 2017) and (Driankov & Saffiotti, 2001) respectively. A somewhat representative sample of more recent research is seen in (Etlik, Korkmaz, Beke, & Kumbasar, 2021), (Rastelli & Peñas, 2015), and (X. Wang, Fu, Ma, & Yang, 2015).

The overarching trend in this research is that the controllers are low fidelity nonlinear input-output mappings that attempt to be logically intuitive using linguistic variables and values to partition the decision space. Doing this makes them good at modeling human perception and easy to tune to replicate expert human operator behavior (W. R. Norris, Zhang, & Sreenivas, 2006) (Zadeh, 1975). These controllers all tend to suffer in terms of performance because they have a limited number of linguistic variables with three or fewer linguistic values to keep their rule-base sizes manageable. A common way control theorists have tried to mitigate the downsides of these fidelity problems while maintaining the performance benefits of fuzzy logic is to integrate fuzzy logic into other control schemes.

One of the most common control schemes is to combine fuzzy and sliding mode control. This combination improves the fidelity of a pure fuzzy control system and adds robustness and multi-objective decision-making capability to a pure sliding mode control system. This combination is accomplished by fuzzifying the output of sliding mode controllers to reduce chatter (Lee Jae-Oh, Han In-Woo, & Lee Jang-Myung, 2011) (Song & Smith, 2000). An application of the fuzzy sliding mode controller can be seen (Hwang, Yang, & Hung, 2018) and (Panda, Das, Subudhi, & Pati, 2020) for an Autonomous Ground Vehicle (AGV) and Autonomous Underwater Vehicles (AUV) respectively, both with considerations for payload. Another advantage of fuzzy is its ability to simplify parameter adjustment when paired with other controllers, as in (H. Wang, Li, Liu, Karkoub, & Zhou, 2020), this was achieved when paired with a sliding mode active disturbance rejection controller for an AUV.

Another common approach is to use fuzzy logic to vary the lookahead distance used on a pure pursuit controller (Shan et al., 2015), which allows the pure pursuit controller to function outside of a fixed linear velocity environment. Various other approaches involve using fuzzy logic to pre-process and filter multiple inputs into a single output that is then used as the input to a different style of controller (de Silva, 1993) (Belorkar & Wong, 2016).

These approaches help smooth the fidelity-related issues associated with fuzzy logic but do not solve the underlying issues associated with the lack of membership functions. The Hierarchical Rule-Base Reduction (HRBR) method presented in Section 4 attempts to solve these issues by enabling an increase in the number of linguistic values without an exponential increase in the related rule-base. When paired with the use of trapezoidal membership functions and the resulting symmetric rule-base, additional robustness and ease of implementation are further achieved.

The Clearpath Jackal UGV (Robotics, 2020) and presented in Figure 1, a skid steer vehicle, was used for experimental validation of the controller as well as a template for related simulation validation. Ensuring the accuracy of the Jackal model in simulation, the variable, nonlinear dynamics of an off-road setting were captured by incorporating a description of tire ground interactions using the rigid tire, nonlinear spring-damper model (Azad & Featherstone,

FIGURE 1 The Clearpath Jackal unmanned ground vehicle (Inc., 2022).

2010), as discussed in Section 3. Additional model considerations are also explored in Section 5.3.

The remainder of the paper is organized as follows. Section 3 presents the dynamic model of the Jackal. Section 4 describes the design of the fuzzy logic controller and the HRBR. Section 5 provides associated simulation and experimental studies. Lastly, Section 6 concludes the paper.

3 | DYNAMICS

3.1 | Forward Dynamics

The forward dynamics presented below appeared originally in (Aguilera-Marinovic, Torres-Torriti, & Auat-Cheein, 2016). Five interconnected bodies were used to represent the vehicle. Each of these bodies had a reference frame. The main body referenced frame F1, and each of the wheels referenced frames $F_2 - F_5$, as seen in Figure 2. Accordingly, these wheels were specified as $i = 2, 3, 4, 5$.

The spatial velocity vector for F1 was given by (1). This vector was comprised of the angular and translational velocities represented by ω and v respectively.

$$\mathbf{v}_1 = \begin{bmatrix} \omega_{1x} & \omega_{1y} & \omega_{1z} & v_{1x} & v_{1y} & v_{1z} \end{bmatrix}^T \tag{1}$$

(2) was the velocity of each of the wheels. The motion transformation from F_1 to F_i was given by ${}^i\mathbf{X}_1$, the subspace matrix of each wheel was $\mathbf{S}_i$, and the angular velocity was $\dot{q}_i$.

$$\mathbf{v}_i = {}^i\mathbf{X}_1\mathbf{v}_1 + \mathbf{S}_i\dot{q}_i \tag{2}$$

The inertia matrix of body i at the body's center of mass (COM) defined $\boldsymbol{I}_i$. The inertia matrix for the main body was given by (3) and for the wheels by (4). In (3), a, b, and c represented the dimensions of the body in x, y, and z. Similarly, in (4), 2r, w, 2r represented the dimensions of the wheels. r was the radius of the wheels, and w expressed the wheels' width.

$$\boldsymbol{I}_1 = \frac{m_1}{12}\begin{bmatrix} b^2 + c^2 & 0 & 0 \\ 0 & a^2 + c^2 & 0 \\ 0 & 0 & a^2 + b^2 \end{bmatrix} \tag{3}$$

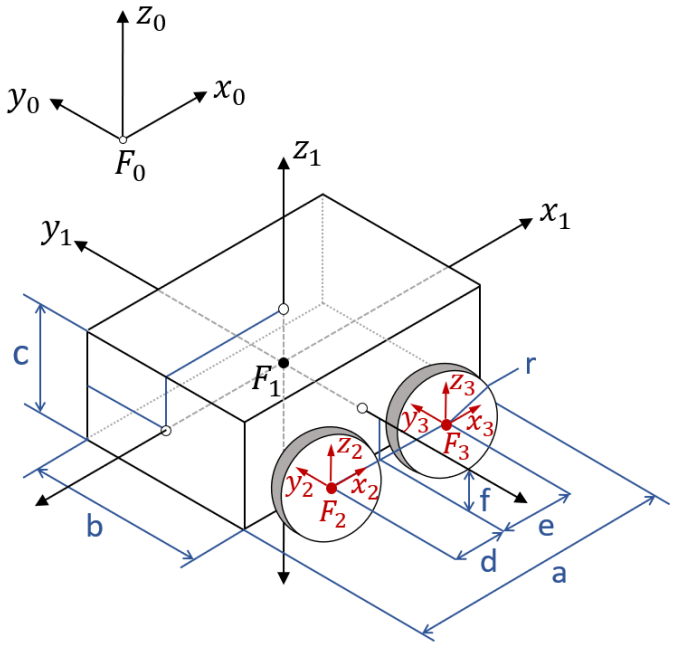


FIGURE 2 The rigid body dynamic model.

$$\boldsymbol{I}_i = \frac{m_i}{12}\begin{bmatrix} 3r^2 + w^2 & 0 & 0 \\ 0 & 6r^2 & 0 \\ 0 & 0 & 3r^2 + w^2 \end{bmatrix} \tag{4}$$

For bodies $n = 1, 2, \cdots, 5$, m_n was the mass of the body. Likewise, the COM location for each body, expressed in body coordinates was given by c_n. Combining the above, the generalized version of the parallel axis theorem for spatial inertia is shown in (5).

$$\mathbf{I}_n = \begin{bmatrix} \boldsymbol{I}_n + m_n \boldsymbol{c}_n \times \boldsymbol{c}_n \times^T & m_n \boldsymbol{c}_n \times \\ m_n \boldsymbol{c}_n \times^T & m_n 1_3 \end{bmatrix} \tag{5}$$

Relative to the main body, (6) represented the apparent inertia of any wheel to the main body.

$$\mathbf{I}_{1/i} = \mathbf{I}_i - \mathbf{I}_i \mathbf{S}_i \left(\mathbf{S}_i^T \mathbf{I}_i \mathbf{S}_i\right)^{-1} \mathbf{S}_i^T \mathbf{I}_i^T, i = 2, 3, \ldots, N \tag{6}$$

The total spatial inertia of the main body was then (7).

$$\mathbf{I}_1' = \mathbf{I}_1 - \sum_{i=2}^{5} \mathbf{I}_{1/i} \tag{7}$$

In addition, (8) determined the force due to the velocity-product.

$$\mathbf{f}_{ic} = \mathbf{v}_i \times^* \mathbf{I}_i \mathbf{v}_i \tag{8}$$

The external spatial force on the main body was evaluated using (9). $\hat{k}_1$ was the unit vector parallel to the z_1 -axis. While the force due to gravity on the main body was f_{1grav} and f^1_{iw} was the reaction force on each wheel.

$$\begin{aligned} \mathbf{f}_{1\text{ext}} &= \begin{bmatrix} n_{1\,\text{ext}\,x} & n_{1\,\text{ext}\,y} & n_{1\,\text{ext}\,z} & f_{1\,\text{ext}_x} & f_{1\,\text{ext}\,y} & f_{1\,\text{ext}\,z} \end{bmatrix}^T \\ &= \begin{bmatrix} \sum_{i=2}^{5} \left(\boldsymbol{r}_i - r\hat{\boldsymbol{k}}_1 \right) \times \boldsymbol{f}^1_{iw} \\ \sum_{i=2}^{5} \boldsymbol{f}^1_{iw} \end{bmatrix} + \mathbf{f}_{1\,\text{grav}} \end{aligned} \tag{9}$$

The inertial acceleration of the main body was given by (10). ${}^1\mathbf{X}^*_i$ represented the force transformation from F_i to F_1.

$$\begin{aligned} \mathbf{a}_1 = & \underbrace{\left(\mathbf{I}'^{-1}_1 \right) \mathbf{f}_{1c} + \left(\mathbf{I}'^{-1}_1 \right) \sum_{i=2}^{N} {}^1\mathbf{X}^*_i \mathbf{f}_{ic}}_{\mathbf{a}_{1c}} \\ & \underbrace{- \left(\mathbf{I}'^{-1}_1 \right) \sum_{i=2}^{N} {}^1\mathbf{X}^*_i \mathbf{f}_{i\ \text{ext}}}_{\mathbf{a}_{1/i}\mathbf{ext}} \underbrace{- \left(\mathbf{I}'^{-1}_1 \right) \mathbf{f}_{1\ \text{ext}}}_{\mathbf{a}_{\text{lext}}} \end{aligned} \tag{10}$$

The angular acceleration of each wheel, $\ddot{q}_i$, was determined using (11). Conversely, the applied torque was given by τ_i and $d_i = {}^i\mathbf{X}_1 \mathbf{a}_1 + \mathbf{v}_i \times \boldsymbol{S}_i \dot{q}_i$.

$$\ddot{q}_i = \left(\tau_i - \mathbf{S}_i^T \mathbf{f}_i - \mathbf{I}_i \mathbf{S}_i \left(d_i \right) \right) \left(\mathbf{S}_i^T \mathbf{I}_i \mathbf{S}_i \right)^{-1} \tag{11}$$

3.2 | Ground Contact

The model presented in the previous sub-section originated from (Aguilera-Marinovic et al., 2016) with the addition of (Azad & Featherstone, 2010). In this model, the vehicle was treated as a rigid body that interacted with a compliant ground. This compliant ground was modeled as a uniform distribution of an infinite number of non-linear spring-damper pairs. Further, these rigid body-ground interactions were represented as a set of discrete contact points, each of which caused the ground to deflect spherically.

The relative modulus of elasticity between the wheel(s) and the ground, E^*, was computed using (12). In (12) the wheel(s) and ground had moduli of elasticity tied to E_w and E_g, and Poisson ratios given by ν_w and ν_g, respectively.

$$\frac{1}{E^*} = \frac{1 - \nu_w^2}{E_w} + \frac{1 - \nu_g^2}{E_g} \tag{12}$$

The stiffness and damping coefficients were defined in (13) where r was the radius of the sphere and α was a

constant.

$$K = -2E^*\sqrt{r}, \qquad D = 4\pi\alpha \tag{13}$$

The normal force from the ground N_k was calculated using (14) at some point k. In (14), δ_k was the penetration distance, $\dot{\delta}_k$ was the penetration velocity, K was the surface stiffness coefficient, and D was the surface damping coefficient. It was assumed that that $\delta_k < 0$, i.e. penetration was into the ground.

$$N_k = \sqrt{-\delta_k}\left[-K\delta_k - D\dot{\delta}_k\right] \tag{14}$$

Correspondingly, the slipping force was given by (15), where μ represented the coefficient of friction.

$$\boldsymbol{f}_{\text{slip}_k} = \mu N_k \tag{15}$$

The stick component of friction between the wheel(s) and ground was evaluated using (16). $\boldsymbol{u}$ was, "the tangential deformation of the ground at the contact point" and V_{sph} was, "the tangential velocity of the bottom point of the sphere." (Azad & Featherstone, 2010)

$$\boldsymbol{f}_{\text{stick}_k} = -K\delta^{\frac{1}{2}}\boldsymbol{u} - D\delta^{\frac{1}{2}}\boldsymbol{V}_{sph} \tag{16}$$

Given the above, the friction force was determined using (17).

$$f_k = \begin{cases} f_{\text{slip}_k}, & \left|f_{\text{slip}_k}\right| < \left|f_{\text{stick}_k}\right| \\ f_{\text{stick}_k}, & \left|f_{\text{slip}_k}\right| \geq \left|f_{\text{stick}_k}\right| \end{cases} \tag{17}$$

4 | CONTROL DESIGN

In fuzzy logic control, crisp inputs, $z \in \mathcal{R}^n$, feed into the input linguistic variables I_n, which are then categorized into in input linguistic values $A_{n,m}$ in a process called fuzzification (Shan et al., 2015). Linguistic values describe their associated variable's performance with descriptors like fast and slow. Membership functions, $\mu_{Z_{n,m}}$, determine what elements comprise the fuzzy set associated with a given linguistic value (Hanumanthakari et al., 2021).

$$\text{IF } I_1 \text{ is } A_{1,2} \text{ AND/OR } I_2 \text{ is } A_{2,5} \text{ THEN } O_1 \text{ is } B_{1,1} \tag{18}$$

After fuzzification, output value membership is determined using IF-THEN rules.(Mamdani & Assilian, 1975-1). This structure is presented in (18) with O_k being the output linguistic variables and $B_{n,m}$ being the output linguistic values. How the AND, OR, and IF-THEN operations interact with the membership functions for the values in the antecedents and consequents varies with implementation.

A Mamdani type implementation using a product AND (t-norm) was used for the proposed controller. The controller also maintained a constant linear velocity with the vehicle's angular velocity as the only output. To calculate

TABLE 1 A summary of the applied hierarchy.

	Metric Used for Fuzzy Relation Control Strategy Classification		
		distErr Target Far	
	distErr Target Near	distErr Line Zero/ Close / Near	distErr Line Far
distErr Target	2	3	3
distErr Line		2	2
$\theta_{Lookahead}$	1		
θ_{Near}		1	
θ_{Far}			1

this crisp output, Center of Mass (CoM) defuzzifcation, (19), was used. In (19), n was the number of membership functions, x_i was the amount of control output for membership function i, and $\mu_c(x_i)$ was the degree of membership in membership function i.

$$x_{CoM} = \frac{\sum_{i=1}^{n} \mu_c\,(x_i)\,(x_i)}{\sum_{i=1}^{n} \mu_c\,(x_i)} \tag{19}$$

It was noteworthy that the controller's rule-base was symmetric, and similar to the rule-base presented in (W. R. Norris et al., 2006). The reason for the symmetry was that the controller was assumed to act with the same magnitude but opposite gains while making right or left turns. Thus, only the left half of the rule-base is provided in Table 2. The proposed fuzzy controller used trapezoidal membership functions, as opposed to the traditional triangular or Gaussian membership functions. This choice is related to a human operator in (W. Norris, Zhang, Sreenivas, & Lopez-Dominguez, 2003).

Furthermore, the use of trapezoidal membership functions reduced bang-bang and improved overall system stability as the flat regions provided a margin of acceptable error in the input, especially around the zero error region. Moreover, using a trapezoid allowed for some of the more desirable traits of a Gaussian function to be captured, (Khairuddin, Hasan, & Hashmani, 2020), in a computationally efficient way.

The proposed controller incorporated more input error functions (5 inputs) than most. This was done while still keeping the rule-base fairly small (40 rules) as compared to the potential hundreds of rules that would result from a standard fuzzy controller with the same linguistic variables and values. This level of fidelity was achieved through use of the Fuzzy Relations Control Strategy (FRCS) introduced in (W. R. Norris, 2001). First, relevant controller linguistic variables, Fuzzy Relations Control Variables (FRCVs), and outputs were established. Then, an FRCS determined the most globally influential FRCVs and put the FRCVs in a hierarchy of influence. This hierarchy was used to divide the operating environment into distinct regions. Following, the relations in the hierarchy/ regions informed a selection of the the rules most influential on state errors. This entire top-down process comprised the HRBR as was introduced in (W. R. Norris, 2001), (W. R. Norris et al., 2006).

Error signal functions, the FRCVs, were minimized when the vehicle was in a specific state with the span of all error states corresponding to all possible vehicle positions and orientations. The FRCVs were: the distance from the vehicle to the target point (distErr Target) Table 3, the minimum distance from the vehicle to the current path segment

TABLE 2 Half of the symmetric fuzzy controller rule-base.

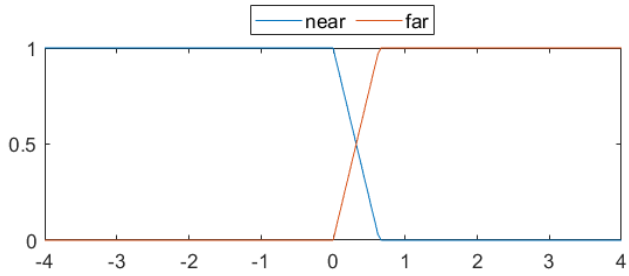
IF					THEN
distErr Target	**distErr Line**	$\theta_{Lookahead}$	θ_{Far}	θ_{Near}	**Control**
near		far left			right 4
near		close left			right 2
near		zero			zero
far	far left		far left		right 4
far	far left		close left		right 1
far	far left		zero		zero
far	far left		close right		left 1
far	far left		far right		left 4
far	near left			far left	right 4
far	near left			near left	right 3
far	near left			zero	right 2
far	near left			near right	right 1
far	near left			far right	left 2
far	close left			far left	right 4
far	close left			near left	right 2
far	close left			zero	right 1
far	close left			near right	zero
far	close left			far right	left 3
far	zero			far left	right 3
far	zero			near left	right 1
far	zero			zero	zero
far	zero			near right	left 1
far	zero			far right	left 3

(distErr Line) Table 4, the angle between the vehicle's heading and the current path segment (θ_{Near}) Table 5, the angle between the vehicle's heading and the next path segment ($\theta_{Lookahead}$) Table 7, and the angle between the vehicle's heading and a point F on the current path segment between the vehicle's projected onto the segment and the next waypoint (θ_{Far}) Table 6.

The distErr target had two associated linguistic values, far and near, partitioned the space into being near the target waypoint or far away from it. distErr Line further partitioned the far from the target space into near the target path and away from the target path see Table 1. distErr Line incorporated seven associated linguistic values to categorize how far away the vehicle was from the target trajectory: far left, near left, close left, zero, close right, near right, and far right. The remaining FRCVs (θ_{Near}), (θ_{Far}), and ($\theta_{Lookahead}$) all used similar linguistic values to qualify the orientation of the vehicle: far left, close/near left, zero, close/near right, and far right.

In Figure 3, A is the waypoint that the vehicle most recently passed, B is the current target waypoint, C is the next waypoint on the trajectory after the current target waypoint, F is the far from trajectory target point, and R is the robot's current position. The distance error signals can be seen in Figure 3 with the aforementioned angular errors

TABLE 3 DistTarget's membership functions.



Linguistic Values	Membership Functions
close	*trap*(-100, -100, 0, 0.6418)
far	*trap*(0, 0.6418, 100, 100)

calculated using (20)-(22).

$$\theta_{near} = \theta_{AB} - \theta_R \tag{20}$$

$$\theta_{Lookahead} = \theta_{BC} - \theta_R \tag{21}$$

$$\theta_{Far} = \theta_{RF} - \theta_R \tag{22}$$

In pure pursuit controllers, path following is achieved by targeting a fixed distance in front of the vehicle's projection onto the trajectory (Lundgren, 2003). The primary downside of this approach occurs when the projection onto the path is very close to the target, but the vehicle has drifted from the path. This scenario leads to the vehicle meeting its completion criteria when it is far away from the target. If this occurs, the vehicle state would then be projected onto the next segment. This could result in an extremely large jump in the projected distance and potentially the skipping of a future waypoint altogether. In obstacle-riddled environments with a small number of navigable paths, obstacle avoidance protocols can lead to such an error cascade.

To avoid this and other problems, the fuzzy controller used (θ_{Far}). When the positional error state was far away from the target trajectory and target point, the controller was designed to orient the vehicle as to minimize (θ_{Far}) and head towards the far from trajectory target point (F in Figure 3). Defining the location of F, as seen in (23), was a non-trivial task as there were benefits and costs to putting it anywhere between the vehicle's projection onto the current path segment and the target point.

$$bF = k * proj_{\vec{AB}} R + (1 - k) \times B \qquad 0 \leq k \leq 1 \tag{23}$$

In order to have the controller approach the trajectory quickly, the value of k had to be close to 1. Figure 4 shows the system approach behavior over a range of values, which were used for tuning purposes to select the desired k. For this controller, a k of 0.95 was selected.

This method to select F provided a harmonious solution that solved several issues. The vehicle aggressively approached the waypoint when segment completion was imminent, while approaching from a more casual angle when the end of the segment was further away. This casual angle decreased the overall completion time for the path,

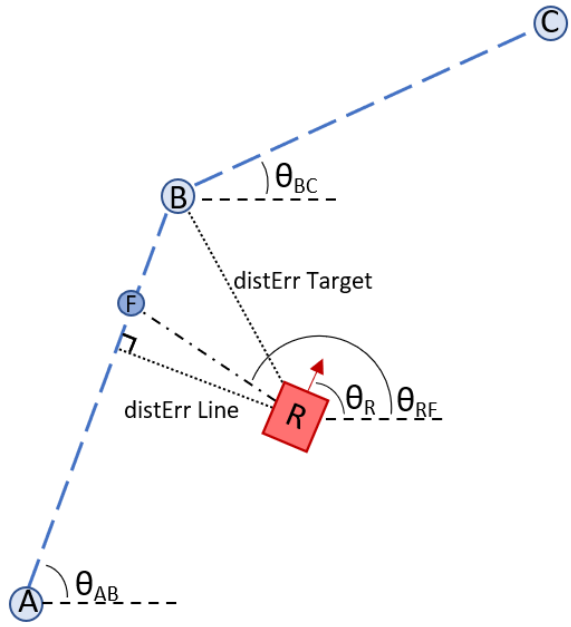


FIGURE 3 Distances and angles relevant to the robot.

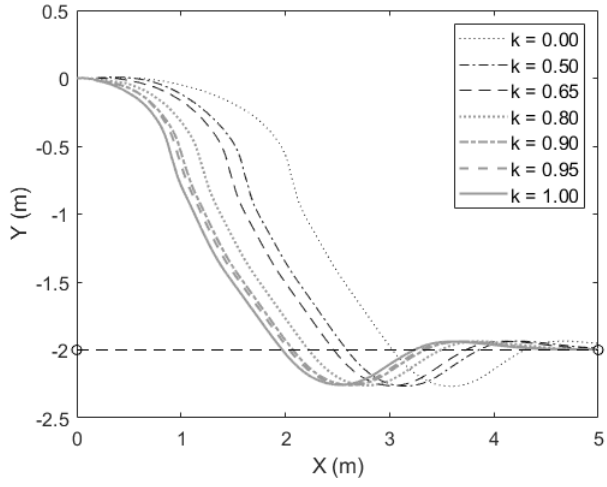
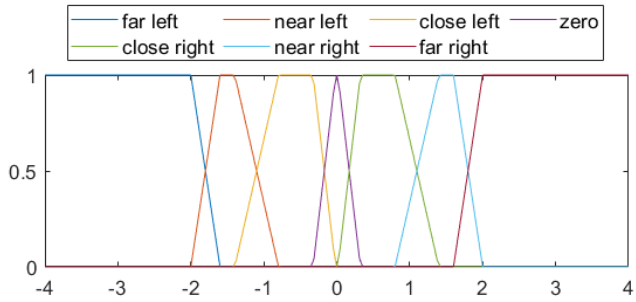


FIGURE 4 The fuzzy system approach behavior over multiple k values on a 5 m target path with a 2 m initial offset.

TABLE 4 DistLine's membership functions.



Linguistic Values	Membership Functions
far left	*trap*(-100, -100, -2, -1.6)
near left	*trap*(-2, -1.6, -1.398, -0.7981)
close left	*trap*(-1.3981, -0.7981, -0.3247, -0.01645)
zero	*trap*(-0.3247, -0.01645, 0.01645, 0.3247)
close right	*trap*(0.01645, 0.3247, 0.7981, 1.3981)
near right	*trap*(0.7981, 1.3981, 1.6, 2)
far right	*trap*(1.6, 2, 100, 100)

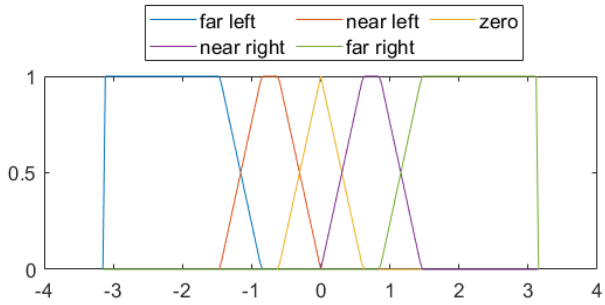
and in cases where the waypoints were far enough apart for the approach angle to be substantially shallow, it would typically be much less important than the path being tightly followed.

When distErr Target was small and in the range of its "near" linguistic value minimizing ($\theta_{Lookahead}$) was the sole control objective. Similarly, minimizing (θ_{Far}) was the sole control objective when distErr Target was in its "far" range and distErr line was in its "far left" range or "far right" range. As a result, both variables had the same linguistic values. They also all mapped to very similar steering control output values such that: far left mapped to right 4, close left mapped to right 2 or 1 respectively, zero mapped to zero, close right mapped to left 2 or 1 respectively, and far right mapped to left 4.

When the vehicle was close to the path but away from the target point, the control objective was multifaceted. It prioritized minimizing (θ_{Near}) while also driving and then maintaining distErr Line to/at zero. This task incorporated five linguistic values from the distErr Line membership functions (near left, close left, zero, close right, and near right), and all five linguistic values from the (θ_{Near}) membership functions (far left, near left, zero, near right, and far right). These were combined to make 25 rules that stabilized the vehicle about its equilibrium point. When the distErr line was zero, the steering control output minimized (θ_{Near}) as follows: far left mapped to right 3, near left mapped to right 1, zero mapped to zero, near right mapped to left 1, and far right mapped to left 3.

In the presented fuzzy system, the output angular velocity setpoint that ranged from $-\omega_{max}$ to ω_{max} also had nine potential linguistic values. Those linguistic values were left 4, left 3, left 2, left 1, zero, right 1, right 2, right 3, and right 4. Unlike the membership functions associated with the input values, the membership functions for the output were triangular, all had the same area, and did not sum to one. The defuzzification process, which combined these control signals, used the CoM approach discussed previously. Each of the output membership functions being triangles with

TABLE 5 Theta near's membership functions.



Linguistic Values	Membership Functions
far left	*trap*(-3.142, -3.142, -1.46, -0.8556)
near left	*trap*(-1.46, -0.8556, -0.6118, -2.447e-05)
zero	*trap*(-0.6118, -2.447e-05, 2.447e-05, 0.6118)
near right	*trap*(2.447e-05, 0.6118, 0.8556, 1.46)
far right	*trap*(0.8556, 1.46, 3.142, 3.142)

the same area meant that the defuzzification would take a weighted average of the peak values of the triangles, with the weights being the percentage that the associated rules were active. The normalized input membership functions had to span all errors to ensure that at all states the vehicle had output function membership. However, the controller performance saw no benefit from having the potential output spanning the space of all possible control inputs.

5 | SIMULATION AND EXPERIMENTAL STUDIES

5.1 | Test Courses

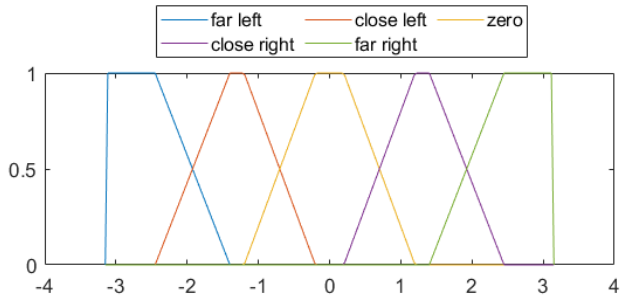
The methodology presented in (W. Norris & Patterson, 2019) for validating controller performance was used for both the fuzzy and pure pursuit controllers. The approach validated controller performance by testing controllers under a set of path conditions that emphasized the efforts of disturbance rejection, phase lag, overshoot, etc.

There are no figures in this section associated with blank versions of each of these test courses. However, each course is shown in Section 5.4 with plots of the vehicle performance overlayed.

Test Course 1 was a figure-eight like path. The at or below minimal turn radii of the circles were used to evaluate the ability of a controller to accommodate the associated steering disturbances as present in the Maximum Error (ME). Meanwhile, the curvature of this design was useful in evaluating path phase lag about the curves which could lead to distance error and thus higher Root Mean Squared Error (RMSE).

The next path, Test Course 2, incorporated above minimum radius turns that were still relatively sharp. These above minimum radius turns allowed for a more accurate assessment of RMSE as a vehicle that could not turn in place could still have zero error. Accordingly, straightaways were paired with these above minimum radius turns to evaluate overshoot. This distinction is more important than it would initially seem as squaring the error term amplifies the errors associated with overshooting. Test Courses 2 also had both right-handed and left-handed turns, thus ensuring

TABLE 6 Theta far's membership functions.



Linguistic Values	Membership Functions
far left	*trap*(-3.142, -3.142, -2.45, -1.4)
close left	*trap*(-2.45, -1.4, -1.2, -0.2)
zero	*trap*(-1.2, -0.2, 0.2, 1.2)
close right	*trap*(0.2, 1.2, 1.4, 2.45)
far right	*trap*(1.4, 2.45, 3.142, 3.142)

that the vehicle operated identically in both directions.

Test Course 3 incorporated both oscillatory turning that invoked phase lag similar to Test Course 1 and the above minimum radius turns paired with straightaways of Test Course 2. Thus, the course allowed for a more holistic examination of the controllers given the factors associated with RMSE and ME discussed above.

5.2 | Pure Pursuit

On each course, the fuzzy controller's performance was compared to that of the classical pure pursuit algorithm implemented in (Samuel, Hussein, & Mohamad, 2016a). This choice was made as pure pursuit is one of the most commonly used waypoint navigation control algorithms and thus made for an ideal baseline controller. For that reason and those discussed in the Introduction, the controller used was the default MATLAB pure pursuit control block (The MathWorks, 2021b). The geometric structure of the controller is presented in Figure 5 with: D being the lookahead distance from the vehicle to the path, and α being the angle between the vehicle's orientation, and r being the radius of the curve that the vehicle R travels along. Moreover, r is computed using D and α with the equation $r = \frac{D}{2\sin(\alpha)}$.

The lookahead distance was chosen to be 0.5 meters by sweeping through potential lookahead values while converging to a straight line with an initial offset. The results of this experiment were run in simulation and can be seen below in Figures 6 and 7. The lookahead value of 0.5 meters was selected because it appeared to have reasonably small overshoots while also having acceptably fast rise times and settling distances. In test cases like the one presented in Figure 6, it made sense to choose a further lookahead point between 0.55 m and 0.65 m because it further reduced overshoot and did not sacrifice much in terms of settling distance. As well, it can be gleaned from 7 that a further lookahead point requires less control action. However, these controllers performed worse in terms of accumulating phase lag. The 0.5 m lookahead controller also performed slightly better than these controllers in terms of minimizing the RMSE over the path; with $\sqrt{mse}$ values of 0.5338 m for 0.5 m and 0.5545 m, 0.5864 m, and

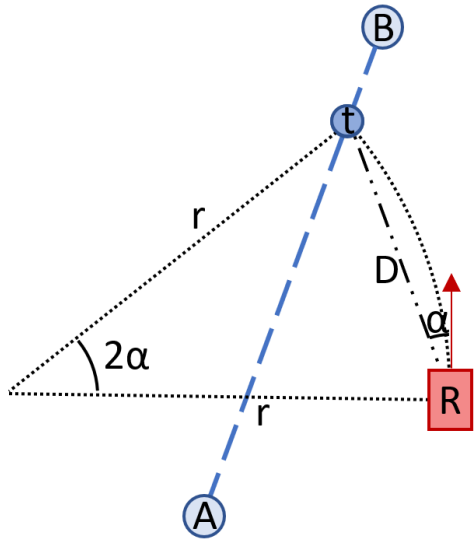


FIGURE 5 The geometry used by pure pursuit to determinate a trajectory.

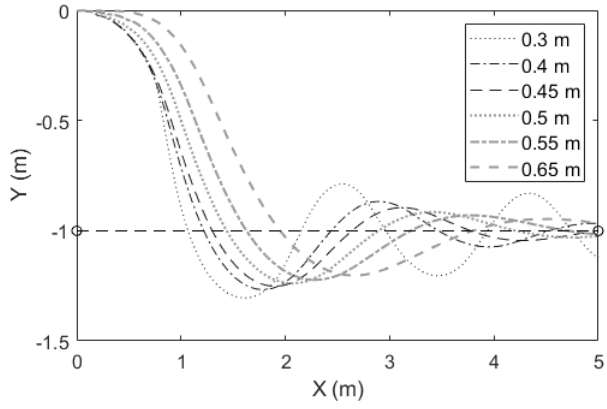
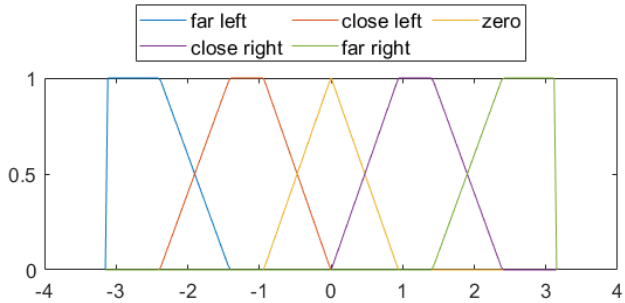


FIGURE 6 The pure pursuit trajectories for multiple lookahead distances on a 5 m target path with a 1 m initial offset.

TABLE 7 Theta lookahead's membership functions.



Linguistic Values	Membership Functions
far left	*trap*(-3.142, -3.142, -2.391, -1.414)
close left	*trap*(-2.391, -1.414, -0.9409, -0.0034)
zero	*trap*(-0.9409, -0.0034, 0.0034, 0.9409)
close right	*trap*(0.0034, 0.9409, 1.414, 2.391)
far right	*trap*(1.414, 2.391, 3.142, 3.142)

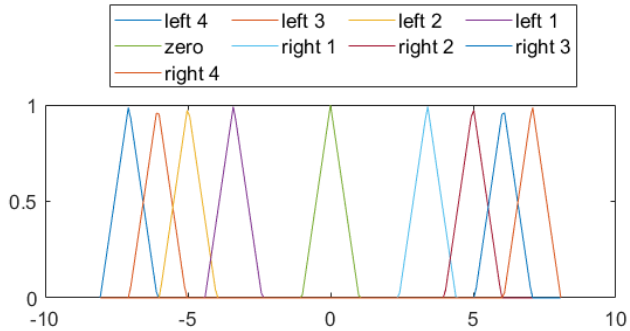
0.5864 m for 0.55 m, 0.60 m and 0.65 m of lookahead distance respectively. There can also be an argument to be made for using slightly smaller lookahead values because they have slightly smaller $\sqrt{mse}$ values; with 0.40 m having had a value of 0.5254 m and 0.45 m having had a value of 0.5270 m. However, on the actual test courses, these controllers performed worse than the controller with a lookahead distance of 0.5 m. This was because the settling distance and overshoot became much more important metrics than rise distance when the vehicle was initialized on the path. If the vehicle was expected to have to regularly converge to paths from far away the benefits from these controllers may outweigh the costs, but this is an edge case in most day-to-day operations.

5.3 | Simulation and Experimental Setup

The metrics used to compare the controllers were the square root of the mean squared distance error with respect to the target trajectory, the maximum distance error with respect to the target trajectory, and the time required to complete the courses. The controller tuning was done by hand and with the aid of automation scripts.

The experimental results were acquired by running the Clearpath Jackal on a lightly worn concrete parking lot. An instance of the Robot Operating System (ROS) ran on the Clearpath Jackal. Using ROS allowed for both sensor data to be sent to and commands to be received from an external laptop. To enable such communication/control the laptop ran MATLAB, the MATLAB ROS Toolbox (The MathWorks, 2021c), the MATLAB Fuzzy Logic Toolbox (The MathWorks, 2021a), and Simulink. In Simulink, a subscriber block subscribed to the position and orientation inputs from the ROS topic '/odometry/filtered.' Next, these inputs were converted into vehicle states and a target trajectory. Those were then fed into the fuzzy controller. The controller proceeded to determine the angular velocity setpoint. Both the predefined linear and controlled angular velocity setpoints were then published to the ROS topic '/cmd_vel' using a publish block. At the same time, the x position, y position, angular velocity, and distErr Line were saved to a matrix in MATLAB.

TABLE 8 Steering membership functions.



Linguistic Values	Membership Functions
left 4	*tri*(-8.0704, -7.0704, -6.0704)
left 3	*tri*(-7.0561, -6.0561, -5.0561)
left 2	*tri*(-5.9934, -4.9934, -3.9934)
left 1	*tri*(-4.3981, -3.3981, -2.3981)
zero	*tri*(-1, 0, 1)
right 1	*tri*(2.3981, 3.3981, 4.3981)
right 2	*tri*(3.9934, 4.9934, 5.9934)
right 3	*tri*(5.0561, 6.0561, 7.0561)
right 4	*tri*(6.0704, 7.0704, 8.0704)

The same data was saved in the simulation where results were acquired using the skid-steer vehicle dynamic model presented in the Dynamics section. Accordingly, the *Spatial_v2* toolbox, as presented in (Featherstone, 2015) and accompanied by (Featherstone, 2008), allowed for implementation of the dynamic model. Related values specific to the Clearpath Jackal can be found in Table 9.

Further, the Simulink portion of the model in the simulation was divided into four major components: the forward dynamics solver, the ground contact model, an external vehicle controller, and an internal vehicle controller.

The forward dynamics solver applied forces/torques to update the vehicle's position and velocity. The ground contact model determined how the ground applied forces and torques back to the vehicle.

The external vehicle controller worked much the same as in the experimental equivalent. It received the vehicle position, orientation, and target path and output the target linear and angular velocities in order to minimize error with respect to the target trajectory.

The internal vehicle controller accepted these velocity targets as inputs and translated them into wheel torques. During the creation of this model, Clearpath was contacted in an attempt to obtain more information about any specifics regarding the internal control processes of the vehicle, but that information was considered a trade secret.

It was thus assumed that the control processes consisted of a high-level and low-level controller. The high-level controller would receive the linear and angular velocity control setpoints and transformed them into lower-level actuator setpoints. The low-level controller would instruct the actuators to hit those set points. This low-level control was

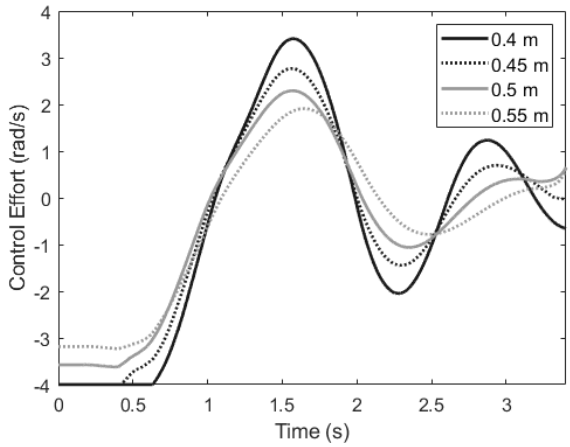


FIGURE 7 The pure pursuit's control actions for multiple lookahead distances on a 3 m stepped course with the step at 1 m.

TABLE 9 The vehicle model parameters of the Jackal.

Property	Chassis	Wheels
Dimension	x: 0.420 m y: 0.310 m z: 0.184 m	r: 0.098 m w: 0.040 m
Offset	x: 0 m y: 0 m z: 0 m	x: 0.131 m y: 0.188 m z: 0.0345 m
Mass	16.52 kg	0.477 kg
Moment of Inertia	$\begin{bmatrix} 0.3136 & -0.002 & 0.0164 \\ -0.0008 & 0.3922 & 0.0009 \\ 0.0164 & 0.0009 & 0.4485 \end{bmatrix}$	$\begin{bmatrix} 0.00116 & 0 & 0 \\ 0 & 0.00229 & 0 \\ 0 & 0 & 0.00116 \end{bmatrix}$

chosen to be a PID controller with an integral windup saturation limit. Related parameters were defined as follows: $K_p = 4$, $K_i = 500$, $K_d = -0.002$, Wheel Velocity Saturation = 2.2 m/s, Torque Saturation = 7 Nm, and Realistic Linear Velocity Factor = 0.9225. Here, the "Realistic Linear Velocity Factor" was used to match the linear velocity the vehicle would actually achieve, given the speed it was commanded to achieve. Furthermore, for ease of implementation, each wheel was modeled as having an associated motor despite the actual vehicle having only one motor for the left two wheels and one motor for the right two wheels.

Additionally, an initialization file was used to create an environment for the vehicle to interact with, which consisted of the ground geometry and ground contact coefficients $K = 1000000$, $D = 1000$, and $\mu = 0.85$. Next, it created a 6 DOF floating base parent link with the physical characteristics of the base of the vehicle. Then, it created wheel models with the physical characteristics of the wheels and linked them to the base with a 1 degree of freedom rotation link. After that, the file defined the contact point locations of the base, which were the corners, and the wheels which were 32 points evenly spaced about the circumference of the wheels. Lastly, the vehicle and wheel initial positions, orientations, and velocities were defined.

To verify the proposed dynamic model and tune any inaccurate parameters a high fidelity motion capture system with 7 motion capture cameras located in the University of Illinois Intelligent Robotics Laboratory were used. These cameras were designed to track the wavelength of light reflected by the silver balls attached to the vehicle. This was done by performing a least-squares regression/triangulation of the position of each of the individual balls, which allowed it to calculate the position of the balls with 1 mm accuracy. The proposed ball configuration was thus deemed sufficient to accurately measure the position/orientation data through differentiation and to develop a polynomial fit to map the wheel velocity setpoint allocation.

The communication delay of the Jackal was also incorporated. To measure the communication time delay, a discontinuity was generate between the vehicle's zero angular velocity and a commanded nonzero angular velocity. The time between the Jackal's localization package recognizing the command and the angular velocity of the wheels changing was then measured. Across several trial, this value averaged out to 0.068 seconds. However, since the simulation was not real-time the delay was increased to 0.075 seconds to reduce time discrepancies between it and the experiment. Due to the uncertainty of the Course Completion Time (CCT) of the simulation, experimental and simulation CCTs were calculated by dividing the total distance traveled by the linear velocity.

For both the simulation and the experiment, the Clearpath Jackal ran at an angular velocity set-point ranging between -4 rad/s and 4 rad/s across all three courses. However, the linear velocity was 2 m/s for the first course and 1.5 m/s for the second and third courses. This gave the vehicle a theoretical minimum turn radius of 0.5 m and 0.375 m respectively.

5.4 | Results

The control efforts, where differences were most visible, for both the simulation and experiment were remarkably similar. As such, the experimental, real-world control efforts are presented for each of the test courses. For a similar reason, only the experimental path plots are presented below. Additionally, a tabulated set of results for all test courses can be found in Table 10. Moreover, in the Course column of Table 10 the numbers 1-3 refer to the specific course, E represents an experimental run, and S represents a run in simulation. For brevity, exact RMSEs, MEs, and CCTs are not discussed as Percent Change (PC) is the most relevant when comparing controllers.

Test Course 1, as seen in Figure 8, and as discussed in Section 5.1 was designed to assesses whether the controllers experienced non-minimal phase lag and layout related steering disturbance. The control efforts can be seen in Figure 9 with the fuzzy controller outperforming the pure pursuit controller in terms of both steering disturbance and phase

TABLE 10 The performance results of the controllers.

Course	Pure Pursuit			Expert Fuzzy System		
	RMSE (m)	Max Error (m)	Time (s)	RMSE (m)	Max Error (m)	Time (s)
1 Sim.	0.1074	0.2561	9.8601	0.0828	0.2385	9.679
1 Exp.	0.1078	2.312	N/A	0.1283	0.1969	9.3785
2S	0.051	0.2289	53.2771	0.0306	0.1541	52.9149
2E	0.0245	0.1122	52.9134	0.0195	0.1269	52.4711
3S	0.0873	0.2237	30.5961	0.0497	0.1163	29.9559
3E	0.0399	0.0962	29.9013	0.0208	0.0942	29.1915

TABLE 11 The performance percent change from the pure pursuit to the fuzzy controller.

Course	RMSE (%)	Max Error (%)	Time (%)
1 Sim.	-22.905	-2.4428	-1.8367
1 Exp.	-88.098	-91.484	N/A
2 Sim.	-40.000	-32.678	-0.6798
2 Exp.	-20.408	13.1020	-0.8359
3 Sim.	-43.070	-48.011	-2.0924
3 Exp.	-47.870	-2.0790	-2.3738

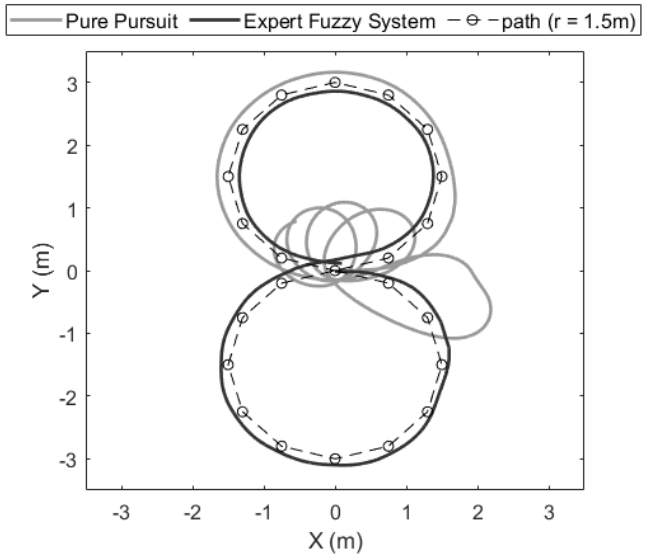


FIGURE 8 The vehicle trajectories on Test Course 1 with the radius of the cirlces comprising.

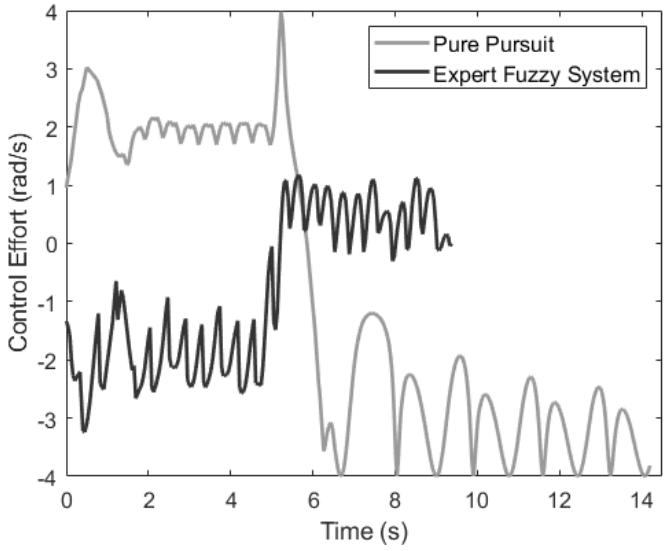


FIGURE 9 The controller setpoints on Test Course 1.

lag.

In terms of metrics, results were exceptional. In simulation, percent changes were -22.905% for RMSE, -2.4428% for ME, and -1.8367% for CCT. The experimental values were larger with a RMSE PC of -88.098%, a ME PC of -91.484%, and no CCT PC as the the pure pursuit had to be manually stopped.

In experiment, the fuzzy controller demonstrated minimal phase lag, despite moderate overshoot and distance from the path. In contrast, the pure pursuit controller experienced a large amount of phase lag after encircling the top circle and was unable to complete the course. While in simulation both controls complete the course similarly, in practise the expert fuzzy system clearly came out ahead of the pure pursuit when large steering disturbance was a factor.

Test Course 2, as seen in Figure 10, and as discussed in Section 5.1 was designed to examine overshoot and whether related instability would be observed.

The associated control efforts can be seen in Figure 11 with the fuzzy controller outperforming the pure pursuit controller in terms of overshoot. The control efforts were otherwise remarkably similar.

Numerically, the controller saw a promising RMSE PC at -40% and ME PC at -32.678% with a marginal decrease in CCT PC at -0.67984% in simulation. While experimentally the RMSE PC was still large but smaller at -20.408%, the ME PC came out to 13.102%. This was the only example of the ME PC being smaller in the pure pursuit of all the trials. Interestingly, the CCT PC still remained fairly low but increased slightly to -0.83589%.

Qualitatively, the results on these trajectories were very similar, outside of the fact the fuzzy controller's control effort quickly settled while the pure pursuit's control effort demonstrated minor overshoot. As well, both controllers remained stable unlike the previous course.

Test Course 3, as seen in Figure 12, and as discussed in Section 5.1 was designed to examine both phase lag and overshoot, as well as if the two combined to create instability in the controllers.

The associated control efforts can be seen in Figure 13 with the fuzzy controller outperforming the pure pursuit

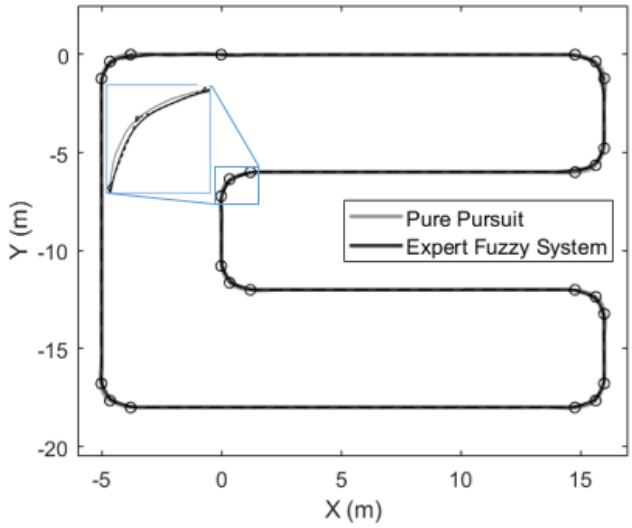


FIGURE 10 The vehicle trajectories on Test Course 2 with a zoom in on one representative turn.

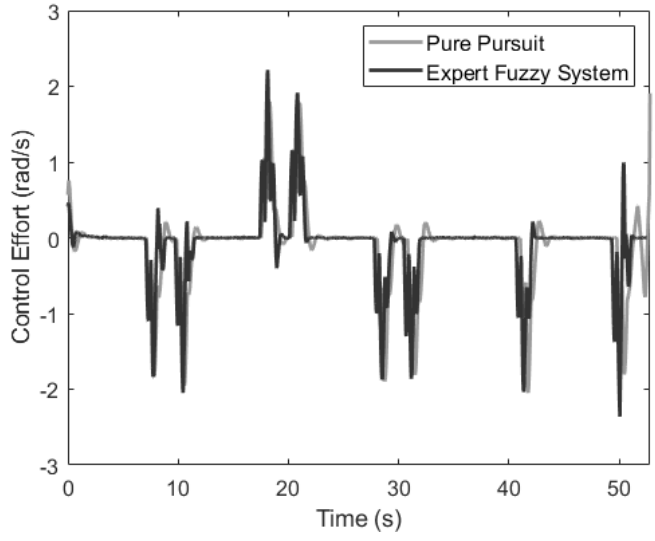


FIGURE 11 The controller setpoints on Test Course 2.

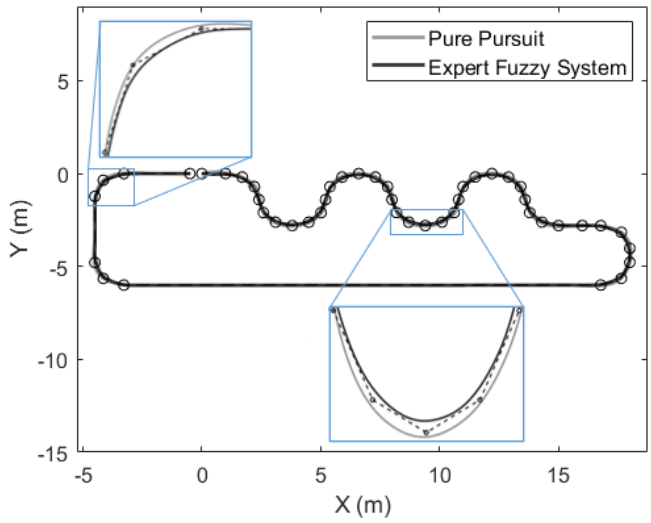


FIGURE 12 The vehicle trajectories on Test Course 3 with zoom ins on two representative turns.

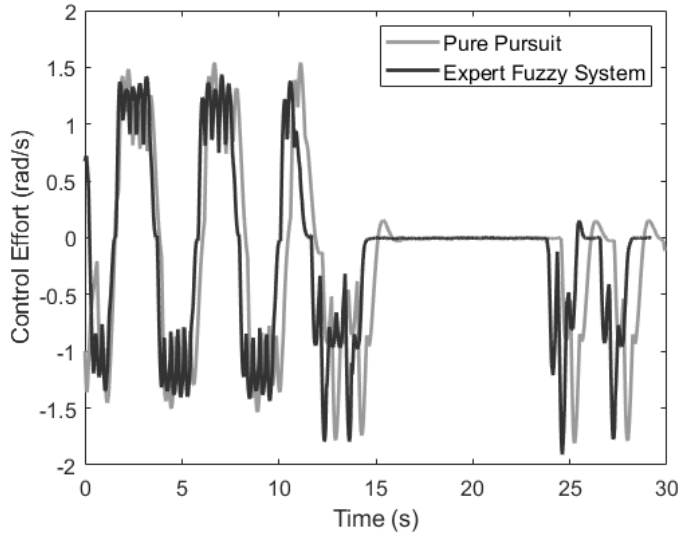


FIGURE 13 The controller setpoints on Test Course 3.

controller in terms of both phase lag and overshoot. As in Test Course 2, the control efforts were otherwise remarkably similar.

In terms of metrics, the fuzzy controller again had a strong showing in simulation with a RMSE PC of -43.070% and a ME PC of -48.011% while the CCT PC was favorable, yet low, at -2.0924%. The experimental results saw a slightly larger RMSE PC at -47.870% than the simulation result. The ME PC was still favorable yet far smaller than the simulation at -2.0790% and the CCT PC of -2.3738% remained at a similarly small value.

For the pure pursuit controller, the phase lag appeared to compound linearly on top of the overshoot as compared to the fuzzy controller. Although, the degree of overshoot for the pure pursuit was similar to what was observed in Test Course 2. As well, both controllers remained stable despite the increased stress on the pure pursuit.

6 | CONCLUSION

In this paper, a novel waypoint navigation control algorithm for a skid-steer vehicle that used the HRBR methodology with the FRCS was presented. This methodology was used to reduce the rule-base of the controller, and as a result the algorithm's computational complexity, while simultaneously increasing the number of input membership functions that the controller could accept from two or three to five. This made for a substantial improvement that better allowed the controller to emulate an expert human operator as it more accurately modeled the complex nonlinear decision-making that humans use. The controller also used trapezoidal membership functions to mitigate the bang-bang effects that traditional controllers, especially fuzzy controllers, can encounter near the zero error state. The symmetric nature of the rule-base further simplified both the rule-base and tuning procedures. As a result of all of the above, the fuzzy controller was seen in the Results section to have outperformed the pure pursuit controller in terms of RMSE (avg. 52.1% improvement), ME (avg. 26.8% improvement), CCT (avg. 1.07% improvement), overshoot, and phase lag. This is noteworthy as pure pursuit controllers are typically used to solve problems of waypoint path following.

It is advised that future work explore the stability of the controller presented in this paper, examine optimal tuning methods, and perform a sensitivity analysis. Moreover, the controller could be paired with obstacle avoidance protocols, additional inputs/outputs, and a second lookahead point could be leveraged to create a second-order approximation of curves. Beyond even skid vehicles this path-following control scheme can be extender to other types of ground vehicles, underwater vehicles, and aerial vehicles given the controllers ability to mimic expert human operator performance. Given the versatile yet highly customizable nature of such a controller, there is even greater potential for the HRBR control type be employed in a wide range of applications far beyond path-following.

References

Aguilera-Marinovic, S., Torres-Torriti, M., & Auat-Cheein, F. (2016, 01). General dynamic model for skid-steer mobile manipulators with wheel–ground interactions. *IEEE/ASME Transactions on Mechatronics*, *22*, 1-1. doi: 10.1109/TMECH.2016.2601308

Azad, M., & Featherstone, R. (2010, December). Modeling the contact between a rolling sphere and a compliant ground plane. In *Proc. australasian conference on robotics and automation, acra 2010*. Brisbane, Australia.

Belorkar, A., & Wong, L. (2016, 12). Gfs: Fuzzy preprocessing for effective gene expression analysis. *BMC Bioinformatics*, *17*, 169-184. doi: 10.1186/s12859-016-1327-8

Bělohlávek, R., Dauben, J., & Klir, G. (2017). *Fuzzy logic and mathematics: A historical perspective*. Oxford University Press. doi: 10.1093/oso/9780190200015.001.0001

Campbell, S. F. (2007). *Steering control of an autonomous ground vehicle with application to the darpa urban challenge* (Unpublished master's thesis). Massachusetts Institute of Technology, Cambridge, MA.

de Silva, C. W. (1993). Hierarchical preprocessing of information in fuzzy logic control applications. In *Proceedings of ieee international conference on control and applications* (p. 457-461 vol.1). doi: 10.1109/CCA.1993.348248

Driankov, D., & Saffiotti, A. (2001). *Fuzzy logic techniques for autonomous vehicle navigation* (Vol. 61). Physica-Verlag. doi: 10.1007/978-3-7908-1835-2

Etlik, U. B., Korkmaz, B., Beke, A., & Kumbasar, T. (2021). A fuzzy logic-based autonomous car control system for the javascript racer game. *Transactions of the Institute of Measurement and Control*, *43*(5), 1028-1038. doi: 10.1177/0142331219889526

Featherstone, R. (2008). *Rigid body dynamics algorithms*. Springer, Boston, MA. doi: 10.1007/978-1-4899-7560-7

Featherstone, R. (2015, June). *Spatial vector and rigid-body dynamics software.* Author's Personal Website. Retrieved from `http://royfeatherstone.org/spatial/v2/index.html`

Ferrari, S., & Stengel, R. F. (2005). Smooth function approximation using neural networks. *IEEE Transactions on Neural Networks*, *16*(1), 24-38.

Hanumanthakari, S., Sekhar, G. C., Behera, H. S., Nayak, J., Naik, B., & Pelusi, D. (2021). *Comparative analysis of different types of membership functions for fuzzy logic controller in direct torque control of induction motor* (1st ed. 2021. ed., Vol. 702). Singapore :: Springer,.

Hellström, T. (2011). *Kinematics equations for differential drive and articulated steering* (Tech. Rep. No. 11.19). Umeå, Sweden: Umeå University, Department of Computing Science.

Hwang, C.-L., Yang, C.-C., & Hung, J. Y. (2018). Path tracking of an autonomous ground vehicle with different payloads by hierarchical improved fuzzy dynamic sliding-mode control. *IEEE Transactions on Fuzzy Systems*, *26*(2), 899-914. doi: 10.1109/TFUZZ.2017.2698370

Inc., C. R. (2022, February). *Jackel unmanned ground vehicle.* Retrieved from `https://clearpathrobotics.com/jackal-small-unmanned-ground-vehicle/`

Jong-Min Yang, & Jong-Hwan Kim. (1999). Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, *15*(3), 578-587. doi: 10.1109/70.768190

Khairuddin, S. H., Hasan, M. H., & Hashmani, M. A. (2020). Integration of cluster centers and gaussian distributions in fuzzy c-means for the construction of trapezoidal membership function. *Mathematics and Statistics*, *8*(5), 559-565.

Khelfi, M., & Abdessameud, A. (2007, October 16). Robust h-infinity trajectory tracking controller for a 6 d.o.f puma 560 robot manipulator. In *Proceedings of ieee international electric machines and drives conference, iemdc 2007* (pp. 88–94). doi: 10.1109/IEMDC.2007.383558

Kuutti, S., Bowden, R., Jin, Y., Barber, P., & Fallah, S. (2019). *A survey of deep learning applications to autonomous vehicle control.*

Lee Jae-Oh, Han In-Woo, & Lee Jang-Myung. (2011). Fuzzy sliding mode control of unicycle robot. In *2011 8th international conference on ubiquitous robots and ambient intelligence (urai)* (p. 521-524).

Lundgren, M. (2003). *Path tracking and obstacle avoidance for a miniature robot* (Unpublished master's thesis). Umeå University, Umeå.

Majid, N., Mohamed, Z., & Basri, A. (2016, 07). Velocity control of a unicycle type of mobile robot using optimal pid controller. *Jurnal Teknologi*, *78*. doi: 10.11113/jt.v78.9415

Mamdani, E., & Assilian, S. (1975-1). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, *7*(1), 1-13.

Murphy, K. N. (1994). Analysis of robotic vehicle steering and controller delay. In *Yuh, volume 5, robotics and manufacturing, pg 631, wailea* (pp. 631–636).

Norris, W., & Patterson, A. (2019, 09). System-level testing and evaluation plan for field robots: A tutorial with test course layouts. *Robotics*, *8*, 83. doi: 10.3390/robotics8040083

Norris, W., Zhang, Q., Sreenivas, R., & Lopez-Dominguez, J. (2003, May/June). A design tool for operator–adaptive steering controllers. *Transactions of the ASAE*, *46*(3), 883.

Norris, W. R. (2001). *A design framework for qualitative human-in-the-loop system development* (Doctoral dissertation, Univ. of Illinois, Urbana-Champaign). Retrieved from `http://www.ece.udel.edu/~qli`

Norris, W. R., Zhang, Q., & Sreenivas, R. S. (2006). Rule-base reduction for a fuzzy human operator performance model. *Applied engineering in agriculture*, *22*(4), 611–618.

Panda, M., Das, B., Subudhi, B., & Pati, B. (2020, 09). Adaptive fuzzy sliding mode formation controller for autonomous underwater vehicles with variable payload. *International Journal of Intelligent Unmanned Systems*, *ahead-of-print*. doi: 10.1108/IJIUS-08-2019-0037

Pentzer, J., Brennan, S., & Reichard, K. (2014). The use of unicycle robot control strategies for skid-steer robots through the icr kinematic mapping. In *2014 ieee/rsj international conference on intelligent robots and systems* (p. 3201-3206).

Rastelli, J. P., & Peñas, M. S. (2015). Fuzzy logic steering control of autonomous vehicles inside roundabouts. *Applied Soft Computing*, *35*, 662-669. Retrieved from `https://www.sciencedirect.com/science/article/pii/S1568494615003798` doi: https://doi.org/10.1016/j.asoc.2015.06.030

Robotics, C. (2020). *Jackal 2020 data sheet.* Kitchener, Canada.

Sachkov, Y. (2019). *Introduction to geometric control.*

Samuel, M., Hussein, M., & Mohamad, M. (2016a). Implementation of the pure pursuit path tracking algorithm. *International Journal of Computer Applications*, *135*, 35-38.

Samuel, M., Hussein, M., & Mohamad, M. (2016b). A review of some pure-pursuit based path tracking techniques for control of autonomous vehicle. *International Journal of Computer Applications*, *135*, 35-38.

Shan, Y., Yang, W., Chen, C., Zhou, J., Zheng, L., & Li, B. (2015). Cf-pursuit: A pursuit method with a clothoid fitting and a fuzzy controller for autonomous vehicles. *International Journal of Advanced Robotic Systems*, *12*(9), 134. doi: 10.5772/61391

Song, F., & Smith, S. (2000). A comparison of sliding mode fuzzy controller and fuzzy sliding mode controller. In *Peachfuzz 2000. 19th international conference of the north american fuzzy information processing society - nafips (cat. no.00th8500)* (p. 480-484). doi: 10.1109/NAFIPS.2000.877478

Tashiro, T. (2013). Vehicle steering control with mpc for target trajectory tracking of autonomous reverse parking. In *2013 ieee international conference on control applications (cca)* (p. 247-251). doi: 10.1109/CCA.2013.6662766

The MathWorks, I. (2021a). Fuzzy logic toolbox [Computer software manual]. Natick, Massachusetts, United State. Retrieved from `https://www.mathworks.com/help/fuzzy/index.html`

The MathWorks, I. (2021b). Pure pursuit [Computer software manual]. Natick, Massachusetts, United State. Retrieved from `https://www.mathworks.com/help/nav/ref/purepursuit.html`

The MathWorks, I. (2021c). Ros toolbox [Computer software manual]. Natick, Massachusetts, United State. Retrieved from `https://www.mathworks.com/help/ros/index.html`

Vinodh Kumar, E., & Jerome, J. (2013). Robust lqr controller design for stabilizing and trajectory tracking of inverted pendulum. *Procedia Engineering*, *64*, 169-178. Retrieved from `https://www.sciencedirect.com/science/article/pii/S1877705813016020` (International Conference on Design and Manufacturing (IConDM2013)) doi: https://doi.org/10.1016/j.proeng.2013.09.088

Wang, H., Li, X., Liu, X., Karkoub, M., & Zhou, L. (2020, 10). Fuzzy sliding mode active disturbance rejection control of an autonomous underwater vehicle-manipulator system. *Journal of Ocean University of China*, *19*, 1081-1093. doi: 10.1007/s11802-020-4250-6

Wang, X., Fu, M., Ma, H., & Yang, Y. (2015). Lateral control of autonomous vehicles based on fuzzy logic. *Control Engineering Practice*, *34*, 1-17. Retrieved from https://www.sciencedirect.com/science/article/pii/S0967066114002342 doi: https://doi.org/10.1016/j.conengprac.2014.09.015

Yi, J., Song, D., Zhang, J., & Goodwin, Z. (2007, 05). Adaptive trajectory tracking control of skid-steered mobile robots. In *Proc. IEEE int. conf. robot. autom. '07* (p. 2605 - 2610). doi: 10.1109/ROBOT.2007.363858

Young, K. D., Utkin, V. I., & Ozguner, U. (1999). A control engineer's guide to sliding mode control. *IEEE Transactions on Control Systems Technology*, *7*(3), 328-342.

Zadeh, L. (1975). The concept of a linguistic variable and its application to approximate reasoning—i. *Information Sciences*, *8*(3), 199-249. Retrieved from https://www.sciencedirect.com/science/article/pii/0020025575900365 doi: https://doi.org/10.1016/0020-0255(75)90036-5