

CoopeRIS: A Framework for the Simulation of Reconfigurable Intelligent Surfaces in Cooperative Driving Environments

Michele Segata^a, Paolo Casari^a, Marios Lestas^b, Alexandros Papadopoulos^c, Dimitrios Tyrovolas^d, Taqwa Saeed^e, George Karagiannidis^{d,f}, Christos Liaskos^c

^a*Department of Information Engineering and Computer Science, University of Trento, Italy*

^b*Department of Electrical Engineering, Frederick University, Cyprus*

^c*Department of Computer Science and Engineering, University of Ioannina, Greece*

^d*Department of Electrical Engineering, Aristotle University of Thessaloniki, Greece*

^e*School of Information Technology, Halmstad University, Sweden*

^f*Applied AI Research Center, Lebanese American University, Lebanon*

Abstract

Future connected vehicles will require high-performance communication technologies for advanced cooperative driving applications such as maneuvering and cooperative perception. mmWave communications can meet the bandwidth requirements of such applications, but the typically harsh propagation conditions of vehicular environments hinder the broad adoption of mmWave devices on cars. Reconfigurable intelligent surfaces (RISs) can help mitigate this problem by enabling the reflection of signals in a configurable direction. In turn, this can result in more stable non-line of sight (NLoS) links whenever a LoS path is not available. RISs have recently gained attention in the vehicular domain but, while providing benefits, they also introduce a lot of research challenges. To measure their effectiveness at scale, it is necessary to develop simulation tools that can reproduce their characteristics with high fidelity and federate them with existing cooperative driving simulation frameworks. In this work we present CoopeRIS, an open-source simulation framework federated within the PLEXE/Veins/SUMO ecosystem, capable of modeling and simulating RIS-based mmWave communications in a vehicular environment. We exploit CoopeRIS to perform an initial feasibility study, highlighting the challenges ahead and the performance RISs need to deliver in order to enable this type of communication. In addition we propose a method to combine multiple RIS configurations into a single one to enable multi-user service delivery, showing its performance via CoopeRIS. The insights we presents within this work show the potential of such simulation framework and thus how the community can build further research work on top if it.

Keywords: Reconfigurable intelligent surfaces, Cooperative driving, Discrete event simulation

1. Introduction

Car accidents and traffic congestion stand out as significant challenges in smart cities, resulting in human casualties and economic losses [1]. Cooperative autonomous vehicles (CAVs) emerge as a promising solution, as they can perceive their surroundings and adapt to varying road conditions, executing driving functions reliably [2]. Nevertheless, the successful deployment of CAVs depends on a robust and high-performance wireless communication infrastructure. Cooperative driving and perception, vehicular edge computing (VEC), and distributed learning are examples of applications requiring such

infrastructure [3, 4].

In this context, mmWave communications are regarded as a high-bandwidth technology [5] that can effectively support cooperative vehicular applications. However, mmWaves are susceptible to significant signal attenuation in the presence of obstructing objects, such as buildings and intersections [6]. The unpredictable occurrence of blockages and the ensuing temporary disconnections may hinder mmWave technology from providing the reliable, high-bandwidth, and low-delay communication that is essential for cooperative vehicular applications.

Programmable wireless environments (PWEs)

have recently emerged as a novel wireless communication paradigm, aiming at transforming wireless propagation into a fully controllable process, and thus mitigate previously unmanageable phenomena [7]. PWEs achieve this by employing RISs to coat planar objects in a given environment, supporting well-defined networking and programming interfaces.

RISs passively manipulate the power, direction, polarization, and phase of impinging waves by adjusting the impedance of their reflecting elements [8]. Common methods to achieve such tuning involve the use of varactors and varistors, or the construction of reflecting elements from suitable materials such as liquid crystals. The ultimate goal is to achieve a desired radiation pattern for reflected signals [9, 10, 11]. The strategic deployment of multiple RISs can be dynamically orchestrated to establish custom end-to-end wireless propagation routes between vehicles. This makes it possible to bypass obstacles and to create virtual LoS links, thereby enhancing the quality and reliability of the channel [12].

Despite the interest on the topic, simulation frameworks that consider RISs, vehicular mobility, and environment characteristics at the same time are not yet available to the community. To overcome this limitation and provide the community with such tools, we develop and publicly release CoopeRIS¹. The framework is built within the popular PLEXE/Veins/SUMO ecosystem [13], supports multiple RISs, and accounts for their structural characteristics, i.e., the number and density of their radiating elements. The paper describes the RIS model (Sect. 3) as well as channel models and the implementation in a discrete event simulation (DES) (Sects. 3.1 and 4.2). It is built on top of our prior work [14, 15], which has been extended by properly formalizing the model, supporting the computation of RIS gains for erroneous transmitter positions, as well as pairing it with a MATLAB model that can be used to check the correctness of the RIS’s configurations.

We show the effectiveness of CoopeRIS via a feasibility study and exploration of the RIS/autonomous driving synergy in the mmWave band, where we

present important insights, relevant for vehicle-to-vehicle communications (V2V) in the design phase (Sect. 7). Finally, we propose an algorithm to combine multiple configurations into a single one. For instance, our proposed method can be used to split an incoming beam and focus it towards multiple directions, thus covering multiple receivers at the same time (Sect. 5). The algorithm has also been implemented and is available in CoopeRIS, showing its potential for the design of future communication technologies for cooperative driving.

2. Related work

Recent research focusing on vehicular communication scenarios pushes RISs as a technology that can potentially preserve reliability under fast mobility [16]. Notable examples include a robust transmission scheme proposed for RIS-assisted vehicular communication [17], investigations into the performance of a vehicular RIS-assisted network in terms of outage probability and security [18, 19], and the optimization of RIS placement to maximize received power in communication scenarios involving a base station and an autonomous vehicle [20]. Additionally, studies have explored communication aspects that affect vehicular networks, such as configuration delays and power consumption [21].

Several of the above works exploit numerical analysis for performance evaluation [17, 22, 18, 19, 20, 23]. Such analyses provide fundamental results, but the rigorous mathematical modeling makes it unfeasible to consider complex scenarios like cooperative driving functionalities, where the behavior of the network influences the dynamics of the vehicles. In this regard, the work in [21] exploits the micro-mobility simulator VISSIM to generate mobility traces and then analyze the RIS-enabled communication environment by means of electromagnetic simulations. Yet, the network simulator has no mean of influencing the mobility simulator.

Some works focus on the development of simulation platforms for RIS systems [24, 25]. The work in [24] develops SimRIS, a MATLAB-based tool that can be used to generate channel matrices given the configuration of the surface and the position of transmitter and receiver node. The authors publicly release the tool as open source, but the tool itself is limited to a single pair of statically positioned transmitter and receiver, supports up to one RIS, and cannot consider transceivers at different locations than those the RIS has been configured for.

¹Note to the reviewers: we will publish the framework if the paper is accepted, after the acceptance notification. We provide the reviewers with a confidential private link they can use to download the software, to prove we have it ready for release: <https://drive.google.com/drive/folders/1-5mbEYVvRKAqxONmKbGW30LAT9zKQU4p?usp=sharing>

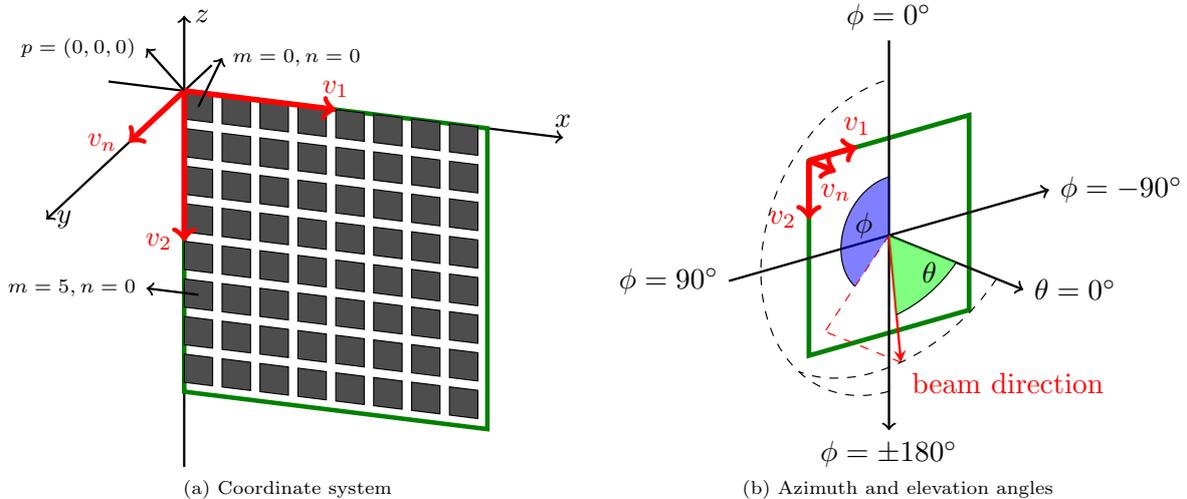


Figure 1: Graphical representation of the coordinate system of the RIS model, together with azimuth and elevation angles. The RIS in the picture is identified by position $p = (0, 0, 0)$ and orientation vectors $v_1 = (1, 0, 0)$, $v_2 = (0, 0, -1)$.

Another example is NYUSIM-RIS [25]. It is based on NYUSIM [26], a mmWave simulator available for both ns-3 and MATLAB. The paper, however, describes only numerical simulations, and NYUSIM is not yet available to the public. Hence, it remains unclear whether with the ns-3 version can actually simulate RISs and vehicular mobility scenarios.

Finally, the work in [27] develops a C++ system-level simulation platform which takes into account the network topology, antenna patterns, network traffic types and service load models, channel measurement and feedback process, as well as resource scheduling and uplink power control. However, the framework does not feature node mobility and it is not yet publicly available.

Last but not least, the performance evaluation of such systems in the real world is not straightforward. The interest by companies like Huawei² and Qualcomm³ is evident, but there are only very few industries that produce prototypes, like ZTE⁴ and Metamaterial⁵, making it difficult to obtain and test

²<https://www.huawei.com/en/huaweitech/future-technologies/terahertz-sensing-communication>, last accessed March 19, 2024.

³<https://www.qualcomm.com/news/onq/2022/11/its-2022-and-were-already-thinking-about-the-evolution-of-5g-advanced-to-6g>, last accessed March 19, 2024.

⁴https://www.zte.com.cn/global/solutions_latest/5g-advanced/ris.html, last accessed March 19, 2024.

⁵<https://metamaterial.com/industries/5g-communications/>, last accessed March 19, 2024.

them.

For the aforementioned reasons we started developing CoopeRIS in our previous work [14, 15]. In this work, we extend the idea further by providing the first release of the software, showing its potential for the analysis of RIS-based cooperative driving systems, as well as the development of new ideas for such type of communications by devising and analyzing a beam splitting algorithm capable of combining multiple RIS configurations into a single one.

3. Reference model

This section describes in detail the reference model, including the coordinate system, angles, the formal description of the RIS, etc. This is in general overlooked in some papers assuming the reader is confident with the subject or that the concept is trivial. Yet, regardless of the expertise of the reader, formally describing the model is fundamental to enable the community to reproduce the results, understand the model, and build on top of existing work. Overlooking these details might lead to ambiguities and to incorrect outcomes. For quick reference and to ease the interpretation of formulas, we list all the mathematical symbols in Tab. 1.

We start by defining the coordinate system, which is assumed to be a three-dimensional, left-handed Cartesian system. This choice relies on the fact that OMNeT++ is indeed left-handed, with the y axis growing towards south. Assuming the same coordinate system eases the implementation.

Table 1: List of mathematical symbols.

Symbol	Meaning
v_1, v_2, v_n	Vectors indicating the orientation of the RIS and its surface normal
m, n	Indexes of reflecting elements in the RIS in the direction of v_2 and v_1 , respectively
ϕ, θ	Azimuth and elevation angles. Different subscripts indicate incidence (i), reflection (r), transmitter (tx), and receiver (rx) angles
λ	Wavelength [Hz]
ρ_λ, N_λ	Number of elements per wavelength [#] and size of the RIS in multiples of λ [m]
d_u, N_{el}	Distance between reflecting elements [m] and total number of reflecting elements [#]
$P_{\phi_{rx}, \theta_{rx}}$	Power emitted towards receivers direction (ϕ_{rx}, θ_{rx}) [W]
$f(\phi, \theta)$	Scattering pattern of a reflective element for direction (ϕ, θ)
$a_{m,n}$	Amplitude gain of the reflective element in position (m, n)
$\Phi_{m,n}, \Theta_{m,n}$	Phase shift of the reflective element in position (m, n) due to coding and due to transmitter and receiver positions, respectively
$A(\theta, d_\theta, d_\phi)$	Area of the spherical element of a unit sphere for elevation θ and for elevation and azimuth angular resolutions d_θ and d_ϕ
G	Normalized gain of the RIS [dBi]
Ω	Set of K configuration matrices to be combined into a single one
Δ	Finite set of discrete values available for configuration, either for amplitudes or phases
Λ	Matrix obtained as a result of combining matrices in Ω into a single one

In the model we identify the surface by three vectors, i.e., p , v_1 , and v_2 . p is the position of the top-left corner of the RIS while v_1 and v_2 define the orientation of the surface. In particular, if we assume the RIS to be placed on the xz plane with the reflective part looking towards the positive y direction, then v_1 would be in the direction of the x axis while v_2 in the direction of the negative z axis. The reflecting side of the surface is identified by the direction of the normal to the surface, computed as the cross product between v_1 and v_2 , i.e., $v_n = v_1 \times v_2$. Fig. 1a shows an example for $p = (0, 0, 0)$, $v_1 = (1, 0, 0)$, and $v_2 = (0, 0, -1)$.

The reflecting elements of the surface are identi-

fied by two indexes m and n : m indexes the elements in the direction of v_2 while n in the direction of v_1 . The reflecting element in position $m = 0, n = 0$ corresponds to the top-left corner of the surface and it is located in position p . In addition, the surface is described by how many reflecting elements can be placed in the space of a wavelength (ρ_λ) and the size of the surface in multiples of the wavelength (N_λ). The distance between the elements is simply $d_u = \frac{\lambda}{\rho_\lambda}$, where λ indicates the wavelength. By default, the model considers a square surface, hence the total number of unit cells is $N_{el} = (\rho_\lambda \cdot N_\lambda)^2$.

Finally, the direction of a beam being, either incident to or reflected by the surface, is identified

$$\Phi_{m,n} = \frac{2\pi d_u}{\lambda} [n(\cos(\phi_r) \sin(\theta_r) - \cos(\phi_i) \sin(\theta_i)) + m(\sin(\phi_r) \sin(\theta_r) - \sin(\phi_i) \sin(\theta_i))] \quad (1)$$

$$\Theta_{m,n} = \frac{2\pi d_u}{\lambda} [n(-\cos(\phi_{rx}) \sin(\theta_{rx}) + \cos(\phi_{tx}) \sin(\theta_{tx})) + m(-\sin(\phi_{rx}) \sin(\theta_{rx}) + \sin(\phi_{tx}) \sin(\theta_{tx}))] \quad (2)$$

$$P_{\phi_{rx}, \theta_{rx}} = \left| \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(\phi_{tx}, \theta_{tx}) f(\phi_{rx}, \theta_{rx}) a_{m,n} e^{-j(\Phi_{m,n} + \Theta_{m,n})} \right|^2, \forall (\phi_{rx}, \theta_{rx}) \quad (3)$$

$$A(\theta, d_\theta, d_\phi) = \begin{cases} d_\phi \left(1 - \cos\left(\frac{d_\theta}{2}\right)\right) & \text{if } \theta = 0 \\ d_\phi \left(\cos\left(\theta - \frac{d_\theta}{2}\right) - \cos\left(\theta + \frac{d_\theta}{2}\right)\right) & \text{if } 0 < \theta < \frac{\pi}{2} \\ d_\phi \left(\cos\left(\theta - \frac{d_\theta}{2}\right)\right) & \text{if } \theta = \frac{\pi}{2} \end{cases} \quad (4)$$

in spherical coordinates by the angles (ϕ, θ) , where ϕ indicates the azimuth and θ the elevation. Fig. 1b depicts an example. The elevation θ is the angle between the beam and the surface normal, so $\theta = 0^\circ$, indicates a normal direction while $\theta = 90^\circ$ a direction parallel to the surface. The azimuth ϕ is the angle measured between the north of the plane of the surface and the projection of the beam on the plane, in a counterclock-wise direction. So $\phi = 0^\circ$ indicates north, $\phi = 90^\circ$ west, $\phi = -90^\circ$ east, and $\phi = \pm 180^\circ$ south.

Next, we describe the model of the RIS in [28] and its implementation. The working principle is to configure the RIS through a procedure called coding. Its objective is finding the right phase shift for each element to focus a signal coming from a given incident angle towards a chosen reflection angle. In particular, let (ϕ_i, θ_i) and (ϕ_r, θ_r) be the intended incident and reflection directions, respectively. Eq. (1) defines the phase to be set for the m, n -th element. Intuitively, Eq. (1) causes each element to compensate for the phase shift of the incoming signal due to the arrival angle, as well as phase shifting the outgoing signal so that the contributions of the elements sum up in-phase towards the desired reflection direction. Notice that, due to hardware limitations, $\Phi_{m,n}$ might be discretized to a set of N_s pre-defined phases, simply choosing the phase that is the closest to the actual value of $\Phi_{m,n}$. In addition, there is the possibility of configuring the reflection amplitude of each single element by setting the value $a_{m,n}$. If not stated otherwise, we assume $a_{m,n} = 1$.

To compute the action of the RIS, i.e., its isotropic gain, we consider the direction of the incoming signal (ϕ_{tx}, θ_{tx}) which depends on the position of the transmitter and the direction towards the receiver (ϕ_{rx}, θ_{rx}) . The effect of the position of transmitter and receiver is an additional phase shift for each of the elements. If the positions of transmitter and receiver match the configuration of the RIS, then the contribution of all elements will be perfectly in-phase and will result in the best gain, otherwise the performance of the RIS will be sub-optimal.

Eq. (2) defines the phase shift of each element due to the position of transmitter and receiver. This is equivalent to Eq. (1), just with inverted signs.

To obtain the isotropic gain of the surface, we first compute the power for a given transmitter direction (ϕ_{tx}, θ_{tx}) and a chosen receiver direction (ϕ_{rx}, θ_{rx}) . This is done by summing the contribution of all the elements given transmitter position, i.e., the phase

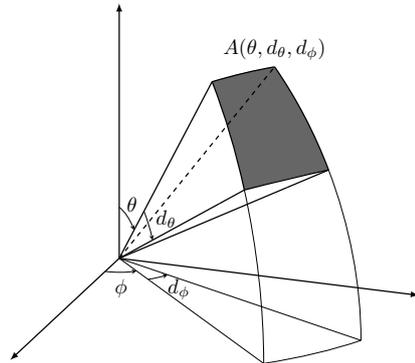


Figure 2: Graphical depiction of a spherical element and the function $A(\theta, d_\theta, d_\phi)$ computing its area.

at the receiver, and then computing its power. We store such power values in a matrix P , with a number of rows and columns that depends on a chosen angular resolution (d_ϕ, d_θ) . With a small abuse of notation, we will address the element of the matrix P storing the power for the receiver direction (ϕ_{rx}, θ_{rx}) as $P_{\phi_{rx}, \theta_{rx}}$, and compute the power as in Eq. (3). In there, $f(\phi, \theta)$ is the scattering pattern of the reflecting elements, which describes how well an element can capture/reflect energy given a certain incident/reflection direction. For simplicity, we assume $f(\phi, \theta) = 1$ but the model can easily be adapted to more realistic scattering patterns.

To compute the final gain, we have to normalize the power over all possible combinations of (ϕ_{rx}, θ_{rx}) with respect to an isotropic radiator. For an ideal isotropic radiator, the power emitted is the same towards all directions, whereas for any real emitter, including RISs, some directions receive more power than others. Still, the sum of the contributions in all directions must be the same as the total emitted power, and we need to make sure about this in the model by means of normalization. Furthermore, given a certain emission direction (θ, ϕ) , the contribution of that direction needs to be weighted by the size of the spherical element in that direction. Fig. 2 shows an example of a spherical element. Clearly, the smaller ϕ the smaller the area of the element, and this needs to be accounted when computing the normalization weight for a given direction.

Starting from the area of a spherical segment between two angles α and β (with $\beta \geq \alpha$), which is defined as

$$2\pi (\cos(\alpha) - \cos(\beta)), \quad (5)$$

we define the normalization weight for a certain ele-

vation θ as in Eq. (4). $A(\theta, d_\theta, d_\phi)$ can be interpreted as the size of the spherical element at elevation θ .

Let ϑ be the vector of all possible values of θ_{rx} given the angular resolution d_θ , i.e., $\vartheta = [0, d_\theta, 2d_\theta, \dots, \frac{\pi}{2}]^T$. We compute the normalized gain as

$$G = P \frac{2\pi}{(P^\top \cdot \mathbf{1})^\top \cdot A(\vartheta, d_\theta, d_\phi)}. \quad (6)$$

In Eq. (6), 2π is the emission area of the isotropic radiator, i.e., the area of a semi-sphere (the RIS does not radiate backwards), whereas the denominator sums all the powers in P weighting each elevation value by the spherical element area.

3.1. Channel model

To derive a comprehensive link budget equation, it is essential to factor in the path loss resulting from multiple reflections, along with the gain introduced by the RIS.

Various path loss models for RIS are found in the literature [29], specifically employing either the product or the sum of distances along the path. The choice between these models depends on the specific scenario being considered. CoopeRIS incorporates both models, offering users the flexibility to easily switch between them based on their scenario of interest.

We begin by describing the channel model referred to as the ‘‘product of distances’’ model. In this model, each hop along the path is treated independently, applying the free space path loss model to each of them. In linear scale, this involves calculating the product of the path losses, effectively determining the total loss by considering the product of the distances. The underlying principle is that the emitter radiates power in all directions, and the free space path loss accounts for the energy reaching the receiver as a function of distance. The RIS then acts as if it was transmitting a new signal, once again in all directions, with a transmission power equal to the received one. The directionality of the RIS is modeled as a gain dependent on the incidence and reflection angles. This process iterates at each RIS until reaching the receiver.

More formally, the ‘‘product of distances’’ model is defined as:

$$P_r = \frac{P_t G_t G_r \prod_{i=1}^{N_{RIS}} G_i^{RIS}}{\prod_{i=1}^{N_{RIS}+1} \left(\frac{4\pi f}{c}\right)^2 d_i^\alpha}, \quad (7)$$

where P_t , G_t , and G_r represent the transmission power, the transmitter antenna gain, and the receiver antenna gain, respectively, f is the frequency and c the speed of light. N_{RIS} is the number of RISs, so $N_{RIS} + 1$ is the number of hops along the full path. G_i^{RIS} is the gain if the i -th RIS, d_i the i -th hop of the path and α the path loss exponent.

The channel model denoted as the ‘‘sum of distances’’ treats the entire path as a single hop. Consequently, the path loss is calculated using the free space formula but considering as total distance the sum of all single hops. As indicated by measurements in [29], this model is applicable only under specific conditions, in particular when operating in the near field of the RIS. This occurs either close to the surface or when the surface is exceptionally large. In such scenario, the received power is defined as

$$P_r = \frac{P_t G_t G_r \prod_{i=1}^{N_{RIS}} G_i^{RIS}}{\left(\frac{4\pi f}{c}\right)^2 \left(\sum_{i=1}^{N_{RIS}+1} d_i\right)^\alpha}. \quad (8)$$

4. Model implementation

This section describes how we integrated RISs within an existing cooperative driving simulation framework. For the sake of clarity, we start by describing the OMNeT++ ecosystem and then how we implemented the RIS model in Sect. 3 within such ecosystem.

4.1. OMNeT++ simulations primer

OMNeT++ is an open source DES software that is freely available for non-commercial usage and has been widely adopted in research studies involving communication networks, including cooperative driving. Discrete event refers to the fact that time is not continuous but it only exists at specific times, that is, when events occur. Components in OMNeT++ are called modules, which basically implement the behavior of a component in C++. In addition to the source implementation, modules are also paired with a network description (NED), a textual file describing properties of modules, their input parameters, etc. An example of a module might be the medium access control (MAC) layer implementation of an IEEE 802.11 wireless interface. The generation and the handling of events by modules is what makes the simulation evolve over time. In particular, modules are inter-connected between them through gates, enabling them to exchange events when this

are generated. Referring to the IEEE 802.11 example, the MAC and the physical layer (PHY) layer modules will be connected via a gate and when the MAC layer has a frame to transmit, it will use such gate to pass it to the PHY, generating a new event in such module. The PHY will pass the frame to the channel which, in turn, will generate beginning of reception events in every PHY module that are in the vicinity of the transmitter.

OMNeT++, however, does not provide any model on its own but only the DES engine plus useful features, such as handling of module parameters, data logging primitives, a graphical user interface (GUI) for debugging, and the like. Models are provided by libraries and frameworks that can be downloaded and compiled. Examples of frameworks include INET⁶, which provides users models of all sort of Internet standards including wired, wireless, and mobile networks, or Simu5G [30], a suite of models to the performance evaluation of 5G networks.

With respect to vehicular networks, the most well-known framework is Veins [31], which in the first place provides an implementation of the IEEE 802.11p standard. In addition, it couples network simulations with the mobility simulator SUMO [32], enabling realistic mobility of network nodes within OMNeT++ and influencing the behavior of vehicles in SUMO, for example upon reception of a message. On top of Veins, PLEXE [13] implements cooperative driving capabilities, such as cooperative adaptive cruise control (CACC) (or platooning) algorithms, engine lag models, and network applications for maneuver management. CoopeRIS, which will be described in the next section, enters the picture as an additional framework for this ecosystem enabling the simulation of RIS-enabled communications.

4.2. CoopeRIS implementation

This section details the integration of the model described in Sect. 3 into CoopeRIS, a DES framework. This framework is built upon the PLEXE, Veins, SUMO, and OMNeT++ ecosystem [13]. An earlier version [14, 15] of the framework only supported a single RIS, the “sum of distances” path loss model, and lacked the ability to configure the metasurface for varying values of N_s , ρ_λ , and N_λ . In addition, the position of the transmitter was always assumed to be the one the RIS was configured for.

Within CoopeRIS, new modules are implemented to facilitate RIS-enabled mmWave channels, alongside a mmWave network interface card (NIC) with RIS capabilities. This NIC is utilized by both standard transceivers and RISs, with the distinction that the NIC of transceivers integrates both a MAC and a PHY layer module, while the NIC of RISs exclusively includes a PHY layer. Currently, the MAC layer of transceivers does not implement any channel access procedure but focuses on data encapsulation and encoding according to the required modulation and coding scheme (MCS). The PHY layer of RISs, on the other hand, is responsible for reflecting and annotating frames with metadata essential for the model. In addition, nodes are assumed to be classic orthogonal frequency division multiplexing (OFDM) transceivers operating at 400 MHz of bandwidth in the mmWave frequency spectrum. This clearly underestimates their performance, but it remains possible to integrate more accurate mmWave models. **An example that could be integrated is the ns-3 mmWave model developed in [33].**

When simulating classical transmissions in a DES, we only consider specific source/destination pairs. However, the application of the gain introduced by a RIS can only be done at the next hop, requiring knowledge of the previous hop, as the gain depends on the incidence and reflection angles. The PHY of a RIS primarily reflects an incoming signal by immediately retransmitting a copy of the incoming signal with zero delay, while adjusting certain properties such as setting a transmission power that matches the incoming one after factoring in the path loss and the gain from the previous RIS (if any). Additionally, the module adds metadata to the frame, such as incidence angles, necessary for computing the gain at the next hop. The module also performs additional operations, including preventing infinite reflections and avoiding the reflection of exceedingly low-power signals for computational efficiency.

In Listing 1 we describe the handling of an incoming frame event in the simulator, which is basically the core part of CoopeRIS. **Listing 1 refers to the routine executed by a module m whenever a frame reception start or end event is notified to such module. Therefore, the notation $m.<property>$ refers to a property of the module, for example its position.** The pseudocode considers a frame to be an object with a set of properties: *power*, *srcType*, *srcPos*, ϕ_{tx} , θ_{tx} , *risList*, and d_{tot} . Power indicates the transmission power prior to the application of the path loss models. This is standard in DES, as

⁶<https://inet.omnetpp.org/>

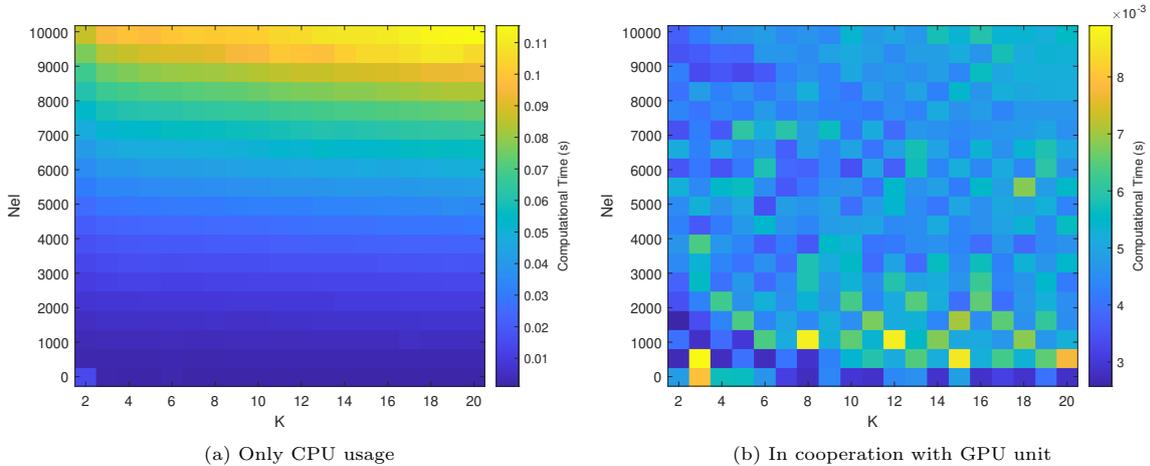


Figure 3: Heatmaps of the required computational time with respect to total RIS elements and number of multiple configurations for the counting method.

the path loss depends on the receiver and so it is applied on reception. The attributes *srcType* and *srcPos* indicate the type (standard transceiver node or RIS) and the position of the transmitter. ϕ_{tx} and θ_{tx} indicate the incidence azimuth and elevation of the transmitter, which are added as metadata by a RIS object before reflecting the signal. In this way the next hop can properly compute the gain using Eq. (6). *risList* indicates the list of RISs that have reflected the signal, while d_{tot} indicates the distance traveled by the signal so far.

The pseudocode in Listing 1 includes two methods: `ONINCOMINGFRAMESTART` and `ONINCOMINGFRAMEEND`. They correspond to the beginning and end of the signal reception events. `ONINCOMINGFRAMESTART` has two parameters, the frame f and the hop distance d . The simulator first checks whether the frame comes from a standard transceiver node or a RIS. If the signal comes from a node, then this is necessarily the first hop, so the power is reduced according to free space path loss. In other words, the first hop is defined at the physical-layer and refers to the first electromagnetic interaction between a non-RIS and a RIS device. If the signal instead comes from a RIS (Line 4) we first compute the amount of loss to be applied depending on the chosen path loss model. With the “product of distances” model, we simply apply the free space path loss computed over the hop distance d , otherwise we apply a loss of $\left(\frac{d_{tot}}{d_{tot}+d}\right)^\alpha$. To compute this factor consider that, under the “sum of distances” model,

the power at the n -th hop should be

$$P_n = P_t \cdot \left(\frac{c}{4\pi f}\right)^2 \cdot \left(\frac{1}{d_1 + \dots + d_n}\right)^\alpha. \quad (9)$$

The attenuation factor between hops $n-1$ and n is then obtained by computing

$$\frac{P_n}{P_{n-1}} = \left(\frac{d_1 + \dots + d_{n-1}}{d_1 + \dots + d_{n-1} + d_n}\right)^\alpha = \left(\frac{d_{tot}}{d_{tot} + d}\right)^\alpha. \quad (10)$$

The next step (Line 9) is to compute the gain provided by the RIS that reflected the signal on the last hop. We compute the reflection azimuth and elevation angles ϕ_{rx} and θ_{rx} and obtain the gain from the far field model, after computing the matrix of gains G as in Eq. (6) given ϕ_{tx} and θ_{tx} . Next (Line 12), we make sure that the RIS does not violate the law of conservation of energy by checking that the gain is at maximum 1. When being very close to the RIS it might happen that the gain is larger than the loss. This can happen when employing the “sum of distances” model because the reference loss $\left(\frac{4\pi f}{c}\right)^2$ is applied only once regardless of the number of reflections (see Eq. (8)). To avoid this, we clip the maximum gain to 1 to make sure that the RIS can emit at most the same amount of energy that it received because we consider a passive metasurface. Finally, we apply the computed loss to the power of the incoming frame.

After applying the gain/path loss models, if the receiving module is a RIS we might need to retransmit

Listing 1 Incoming signal event handling. m represents the module executing the code, f is the incoming frame, d is the hop distance travelled by the frame.

```

1: procedure ONINCOMINGFRAMESTART( $f, d$ )
2:   if  $f.srcType == \text{NODE}$  then  $\triangleright$  frame from a node
3:      $loss = \left(\frac{c}{4\pi f}\right)^2 \frac{1}{d^\alpha}$ 
4:   else  $\triangleright$  frame from a RIS
5:     if use product of distances model then
6:        $loss = \left(\frac{c}{4\pi f}\right)^2 \frac{1}{d^\alpha}$ 
7:     else
8:        $loss = \left(\frac{d_{tot}}{d_{tot}+d}\right)^\alpha$ 
9:      $(\phi_{rx}, \theta_{rx}) = \text{angles}(f.srcPos, m.pos)$ 
10:    compute matrix  $G$  for  $f.\phi_{tx}$  and  $f.\theta_{tx}$   $\triangleright$  Eq. (6)
11:     $gain = G_{\phi_{rx}, \theta_{rx}}$ 
12:     $loss = \min(1, loss \cdot gain)$ 
13:     $f.power = f.power \times loss$ 
14:    if  $m.isRIS$  then  $\triangleright$  this module is a RIS
15:      if  $f.power > \delta$  and  $m \notin f.risList$  then
16:         $f.risList = f.risList \cup \{m\}$ 
17:         $f.d_{tot} = f.d_{tot} + d$ 
18:         $(\phi_{tx}, \theta_{tx}) = \text{angles}(f.srcPos, m.pos)$ 
19:         $f.\phi_{tx} = \phi_{tx}$ 
20:         $f.\theta_{tx} = \theta_{tx}$ 
21:         $f.srcType = \text{RIS}$ 
22:         $\text{transmit}(f)$ 
23: procedure ONINCOMINGFRAMEEND( $f$ )
24:   if not  $m.isRIS$  then
25:     if  $\text{attemptDecoding}(f) == \text{SUCCESS}$  then
26:        $\text{sendToMAC}(f)$ 

```

the signal (Line 14). This only occurs if the received power is larger than a certain threshold δ . If the power is too low, there is no chance receivers will be able to decode the frame after reflecting it, regardless of the gain of the RIS (again, the metasurface we consider is passive). This is common practice in DES and enhances the efficiency of the simulator. The threshold needs to be chosen properly to avoid underestimating interference in multi-access scenarios. An additional condition the simulator checks is that the current RIS has not already reflected such signal before, to avoid infinite reflections. If both conditions are met, we add the RISs to the list of surfaces that have reflected the signal and we update the total distance travelled. Finally, we compute and add the incidence azimuth and elevation ϕ_{tx} and θ_{tx} that will be necessary for computing the RIS gain at the next hop, set the source type, and retransmit the frame.

It is important to notice that here there is no decoding attempt as we are dealing with the beginning of the incoming signal. Decoding is attempted instead at the end of the reception, i.e., when the ONINCOMINGFRAMEEND event is triggered. This is only attempted by standard transceiver nodes and, if successful, the received frame is sent up the

protocol stack.

5. Multi-user phase/amplitude matrices

In this section, we propose and describe an algorithm for combining multiple RIS configurations into a single one. This is particularly useful when individual RIS units serve multiple distinct users, each with unique requirements. For instance, it can facilitate beam splitting, i.e., redirecting an incoming signal toward multiple receivers instead of a single one. The algorithm is not limited to this example, as it can combine any set of configurations into a single one.

Let Ω be the set of K matrices (i.e., configurations) to be combined. The configurations primarily comprise phases and amplitudes. Therefore, the combination of multiple configurations into one is specifically described for the phase component. The same procedure applies to amplitudes as well.

The straightforward method to derive the common RIS configuration Λ is to define it as the average of the multiple configurations intended for combination. Defining Δ as the set of discrete values available, the averaging procedure can be defined as

$$\Lambda_{m,n} = \arg \min_{f \in \Delta} d \left(f, \frac{1}{K} \sum_{k=1}^K \Omega_{m,n}^k \right), \forall m, n, \quad (11)$$

where d is a generic function computing the distance between the two arguments. As averaging can produce values not within the set Δ , Eq. (11) picks the value in Δ that is closer to the average.

In addition to the averaging approach, we also propose the counting method. The primary logic behind this approach is to set each RIS element value equal to the most frequently occurring discretized value among the ones in the configurations to combine. Specifically, the counting function is defined as

$$C_{m,n}(x) = \sum_{k=1}^K [x = \Omega_{m,n}^k], \quad (12)$$

where $[P]$ denotes the Iverson bracket, i.e., 1 if P is true and 0 otherwise. By exploiting Eq. (12), we define the counting method as

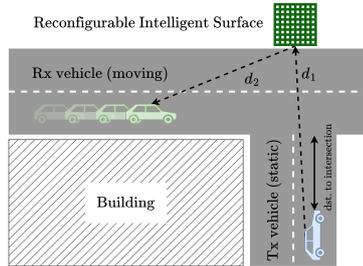
$$\Lambda_{m,n} = \arg \max_{f \in \Delta} C_{m,n}(f), \forall m, n. \quad (13)$$

To break ties in Eq. (13), values are picked at random among the most frequent ones.

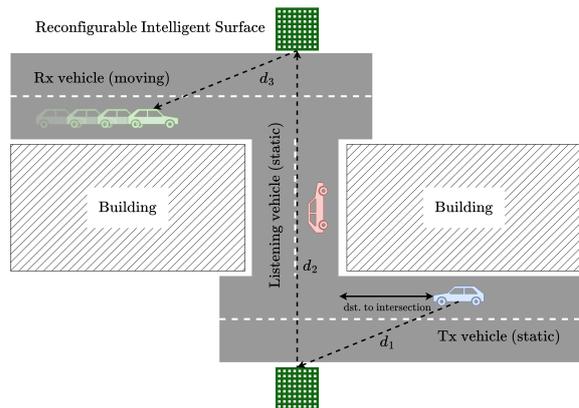
The computational time required by the counting method to derive the common RIS configuration is a crucial aspect to consider. This duration contributes to the overall system response time, making it crucial to minimize its impact on the system’s efficiency. The computational time depends on two primary factors: the size of the RIS and the number of configurations in K that should be combined into a single one. Fig. 3 illustrates, the required computational time to determine the amplitude with respect to the total RIS elements and the number of configurations K for the counting method. The results regarding the computational time of the average method are not presented since this method lacks significantly in terms of performance, as we show in Sect. 7.3. The computational time relative to the two methods is comparable, and with the proper use of the GPU unit, no difference is observed. The measurements are performed using a CPU, specifically, a 13th Gen Intel(R) Core (TM) i9-13900F, and in the other case, in cooperation with a GPU unit, specifically, a NVIDIA GeForce RTX 4090.

In the CPU-only usage scenario, when N_{el} is set to 625 and 2500, the computational time remains between 10 ms and 20 ms across all observed K values. However, for $N_{el}=10\,000$, the required time is more than 100 ms. On the contrary, by exploiting the GPU unit we can reduce such values by up to 13 times. The maximum computational time observed does not exceed 8 ms for any combinations of K and N_{el} . This clearly demonstrates that the proposed algorithm is highly practical for determining RIS configurations to efficiently serve multiple users in realistic scenarios, particularly in terms of computational requirements. Notice that values of N_{el} as large as 10 000 are not unrealistic, as there exist prototypes of this size [34].

All the above measurements have been extracted from the MATLAB environment. The required configurations are assumed to exist and are directly fed into the algorithm as inputs without involving the calculation of RIS configurations in the computational time. Moreover, optimizations have been implemented to enable parallelization in both CPU and GPU cases. Specifically, in the GPU scenario, efforts have been made to minimize data transfer from the CPU to the GPU, enhancing overall performance.



(a) T-intersection scenario



(b) Z-intersection scenario

Figure 4: Scenarios implemented in the simulator.

6. Simulation scenarios

To assess the feasibility of RIS-enabled V2V communications and demonstrate the potential of CooperIS in designing and evaluating such systems, we formulate two scenarios, both featuring intersections with building-induced shadowing. One scenario involves a T-shaped intersection with a single RIS, whereas the other incorporates two intersections with two reflections, such that the central urban channel can be properly connected to the outgoing road legs only via two RISs, as illustrated in Fig. 4. The latter scenario is referred to as the Z-shaped scenario, and the length of its vertical road leg is 60 m. The RISs are positioned directly at the intersection, 10 m above the ground. We configure the road scenario in SUMO using Veins as a bridge between SUMO and OMNeT++. We then exploit PLEXE to control the mobility of the vehicles and, finally, CooperIS as the tool to simulate the RIS-enabled communication between nodes.

In both cases, we examine a static transmitter located at varying distances from the intersection, while the receiver approaches the intersection. The

Table 2: Simulation parameters.

Parameter	Value
Path loss model	Free space ($\alpha = 2.0$), with sum and product of distances
Shadowing model	Simple obstacle shadowing [35]
RIS gain G	Far field model in Eq. (6)
Antenna gains G_t and G_r	1 (isotropic radiator)
Frequency	25 GHz
Transmit power	30 dBm
Transmitter distance to intersection	0 m, 5 m, 10 m, 15 m, 20 m, 25 m, 50 m, 75 m, 100 m, 150 m and 200 m
Coding states (N_s)	8
Elements per λ (ρ_λ)	5
Angular resolutions d_ϕ, d_θ	1°
RIS N_λ (tracking)	5, 10, 20 λ
N_{el} (tracking)	625, 2500, 10000
RIS N_λ (no tracking)	5, 6, 7, 8, 9, 10, 12, 15, 18, 20 λ
N_{el} (no tracking)	625, 900, 1225, 1600, 2025, 2500, 3600, 5625, 8100, 10000

RIS is coded by setting as the incidence angles the ones corresponding to the static transmitter’s location. In the T scenario, the RIS directs the reflected signal toward the receiver. Conversely, in the Z scenario, the first RIS directs the signal towards the second RIS, and the incidence angle of the second RIS is configured to focus the beam from the first RIS and reflect the signal towards the receiver. For each scenario, we consider two sub-scenarios: “tracking” and “no tracking”.

In the tracking scenario, the RIS dynamically tracks the receiver, reconfiguring itself for each signal to be reflected. We assume the RIS to have an “oracle”, so it always knows where the receiving vehicle is located with full accuracy. In a realistic setting, there should be a control unit capable of tracking the vehicle. However, we disregard this aspect in this first study, and rather leave it for future research to design algorithms that tackle such challenge. In the no-tracking scenario, the RIS is statically pre-configured to direct the reflected signal 50 m into the intersection. In addition, within the same scenario, we consider a beam splitting configuration where the RIS directs the reflected signal 50 m and 200 m into the intersection. The T and Z scenarios allow us to explore the impact of single and multiple reflections, whereas the sub-scenarios help assess the necessity of perfect receiver tracking and whether a static configuration suffices, in addition to showcasing the combination algorithm described in Sect. 5.

In the simulation, we vary the RIS size, resulting in different total numbers of elements. In the

tracking scenario, we consider 625, 2500, and 10 000 elements. In the no-tracking scenario, we use the same range for the number of elements but with higher granularity.

Lastly, in the Z scenario, we introduce an additional receiving vehicle located at the middle of the vertical road. This enables us to assess whether multiple receivers can still receive the transmitter’s signal without requiring to reconfigure the RIS. This is particularly interesting in VEC scenarios where data transfer is required across multiple computing nodes, represented here by vehicles. Tab. 2 provides a summary of simulation parameters.

7. Results and analysis

7.1. Impact of the number of element in a RIS

We start by presenting the far field patterns of the RIS model for varying numbers of elements (N_{el}). We obtain these results by using the C++ module for Eq. (6) implemented in CooperRIS, which can be used as a standalone module as well. This analysis shows how N_{el} influences the gain of the metasurface and the characteristics of its beam pattern, providing the background to understand the results in the next sections.

For our analysis, we configure the metasurface for $\phi_r = -45^\circ, \theta_r = 45^\circ, \phi_i = 0^\circ, \theta_i = 0^\circ, N_s = 8$, and for N_{el} values of 625, 2500, 10 000. The transmitter is assumed to be in the correct position, so $\phi_{tx} = \phi_i$ and $\theta_{tx} = \theta_i$. The outcome, depicted in Fig. 5, shows the gain for different N_{el} values in both linear and logarithmic scales.

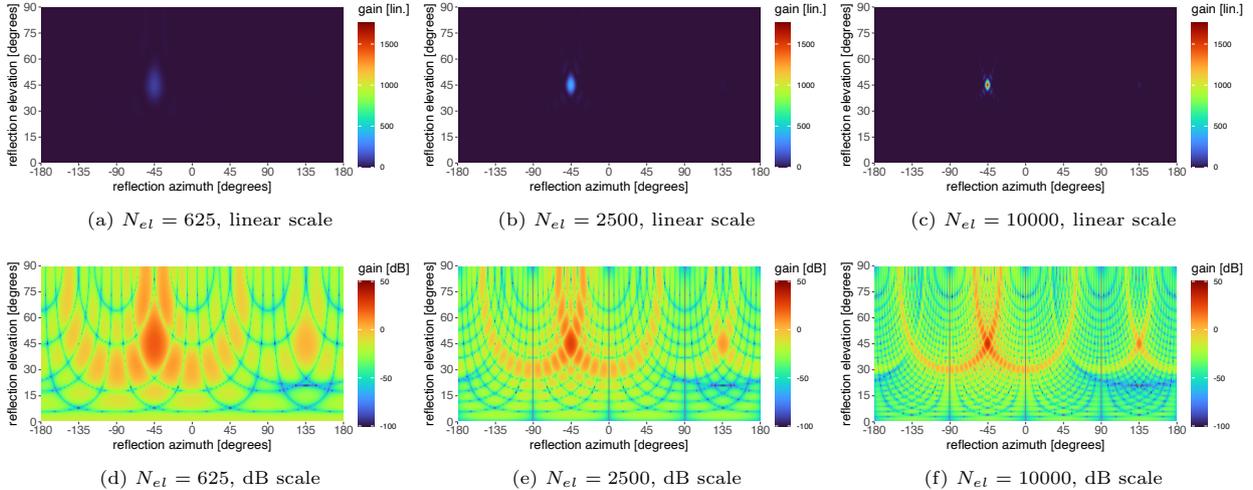


Figure 5: Far field patterns for different number of elements (increasing N_{el}). The RIS is configured for the reflection ($\phi_r = -45^\circ, \theta_r = 45^\circ, \phi_i = 0^\circ, \theta_i = 0^\circ$) and $\phi_{tx} = \phi_i, \theta_{tx} = \theta_i$.

In the linear scale results (Figs. 5a to 5c), metasurfaces with a higher number of elements concentrate more energy towards the expected direction of reflection. This is apparent from the reduction of the beam size and the increase in its maximum gain. It is important to note that determining whether a configuration is better than another depends on the desired balance between broader coverage and higher gain, influenced by specific application requirements and the reference scenario (refer to the analysis in the upcoming sections). Additionally, the actual coverage area depends on the distance from the metasurface; a narrow beam can still cover a substantial portion of the road if it points towards a sufficiently distant location.

The logarithmic scale results (Figs. 5d to 5f) provide a quantitative analysis of far field beam pattern properties. The reader should be aware that the shape of the patterns is affected by “cartographic distortion”, a consequence of mapping a half-sphere onto a flat surface.⁷ Notably, the number of peaks and troughs in the far field pattern increases with the number of elements: this result will be necessary for interpreting the graphs in the following sections.

Finally, we highlight the maximum gains for each configuration, which are difficult to obtain visually. Specifically, the maximum gains are approximately

20 dB, 26 dB, and 32 dB for $N_{el} = 625, 2500,$ and $10\,000$, respectively. These values, while seeming large, are necessary to compensate for the substantial path loss affecting mmWave signals. Moreover, it is worth noting that even better performance can be achieved using directional transmissions, as demonstrated in [36].

7.2. Impact of transmitter position

The model we considered in our past work [15] assumed the transmitter to be always located in the right position, i.e., $(\phi_{tx}, \theta_{tx}) = (\phi_i, \theta_i)$. In here we extend the model to remove such simplifying assumption and show how the model properly accounts for transmitter position. In particular, exploiting the C++ module as in Sect. 7.1, we configure the RIS for a normal incidence ($\phi_i = \theta_i = 0^\circ$) and a reflection direction of $\phi_r = -45^\circ$ and $\theta_r = 45^\circ$ degree. We position the RIS at $p = (0, 0, 0)$, with $v_1 = (1, 0, 0)$ and $v_2 = (0, 0, -1)$. We consider the transmitter to be located at position $(x, 10\text{ m}, 0\text{ m})$ and compute the far-field pattern for different values of x . The position of the transmitter matches the RIS configuration when $x = 0\text{ m}$.

Fig. 6 shows the results for different transmitter positions. Starting from the top-left corner (Fig. 6a), where the transmitter is 25 meters west of its ideal position, we can see that the RIS focuses the reflected beam in a completely different direction, i.e., 90° away from the receiver direction (on the azimuthal plane). As we move the transmitter closer

⁷For instance, this means that the gain at zero reflection elevation is the same irrespective of the reflection azimuthal angle, in the same way as Antarctica is spread across the lower horizontal side of a two-dimensional map of the Earth.

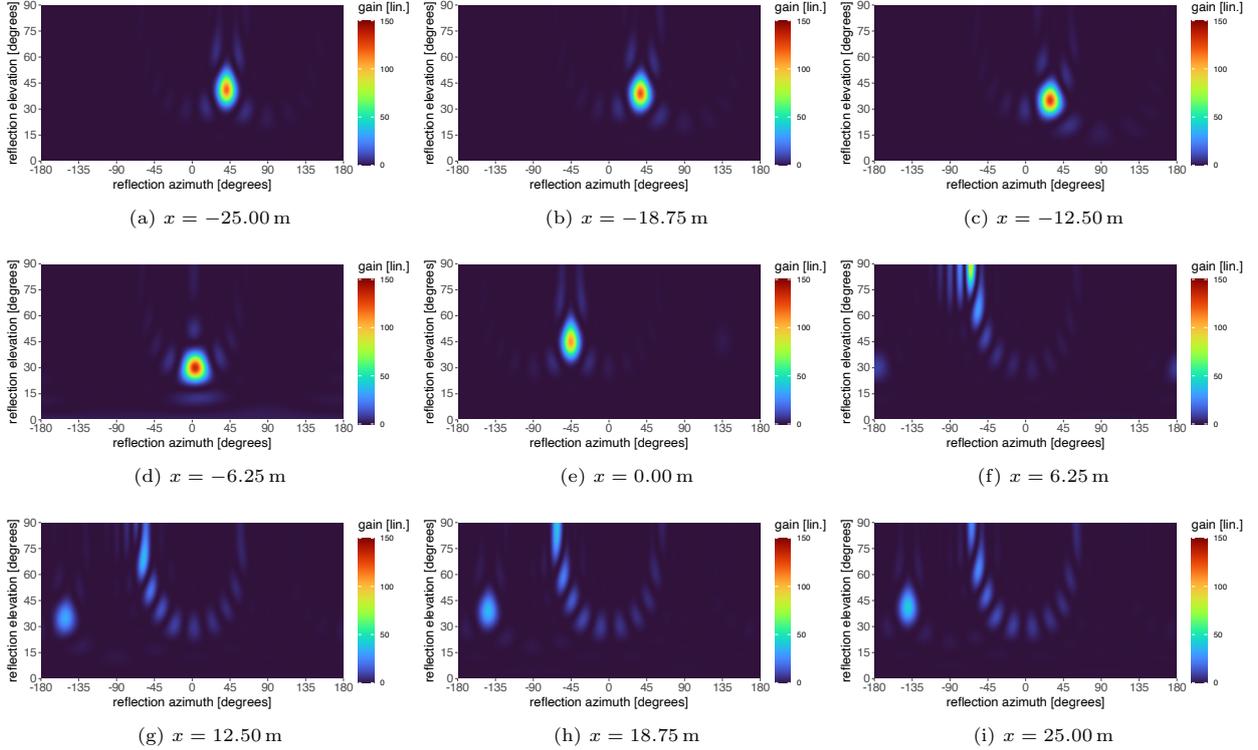


Figure 6: Far field patterns for different positions of the transmitter ($x, 10 \text{ m}, 0 \text{ m}$). The RIS is configured for the reflection ($\phi_r = -45^\circ, \theta_r = 45^\circ, \phi_i = 0^\circ, \theta_i = 0^\circ$). $p = (0, 0, 0)$, $v_1 = (1, 0, 0)$, $v_2 = (0, 0, -1)$.

and closer to its ideal position, the reflection direction converges towards the ideal one but still, for $x = -6.25 \text{ m}$ (Fig. 6d) the reflected beam azimuth is roughly 45° off. When the transmitter moves past the ideal position, the beam gets first focused towards a wrong elevation angle pointing parallel to the surface. As it moves further, the beam gets focused towards an azimuth of -135° . In addition, the surface scatters some energy towards some side lobes, forming a “necklace-shaped” pattern.

The reflection direction errors seem dramatic but we are considering the extreme case where the transmitter is relatively close to the surface, which means that even small changes in its position reflect into large changes of the incidence angle. In this specific example, a change in the x position from -25 m to 25 m translates into incidence azimuth/elevation pairs of roughly $(90^\circ, 68^\circ)$ and $(-90^\circ, 68^\circ)$, respectively, so a total range of 136° for the elevation θ_{tx} . For a more distant transmitter, e.g., with coordinates $(x, 100 \text{ m}, 0 \text{ m})$, the same x range would result in an incidence between $(90^\circ, 14^\circ)$ and $(-90^\circ, 14^\circ)$, so just 28° of elevation change. [This case is shown](#)

in Fig. 7, where the model still shows a pointing error but the beam direction is still relative close to the intended one. The results still showcase the capability of the model to account for positioning errors, which are fundamental in the evaluation of the communication performance in the presence of tracking errors and for the evaluation of tracking techniques as well.

7.3. Combining different configurations

In this section we evaluate the performance of the coding combiner procedure in Sect. 5. In particular, we show the far field patterns for three different configurations. In all of them, we set $\theta_r = 45^\circ$, $\theta_r = 0^\circ$, and $\theta_i = 0^\circ$. As we set the same incidence for all configurations, we are showing the results for a beam-splitting setup. We split the beam towards different reflection azimuths, i.e., $(-45^\circ, 45^\circ)$, $(-45^\circ, 45^\circ, 135^\circ)$, and $(-45^\circ, 45^\circ, 130^\circ)$, testing both the average and the counting strategies.

Fig. 8 shows the resulting far field patterns. The first outcome, seen by comparing the left and the right column, is that the average strategy results

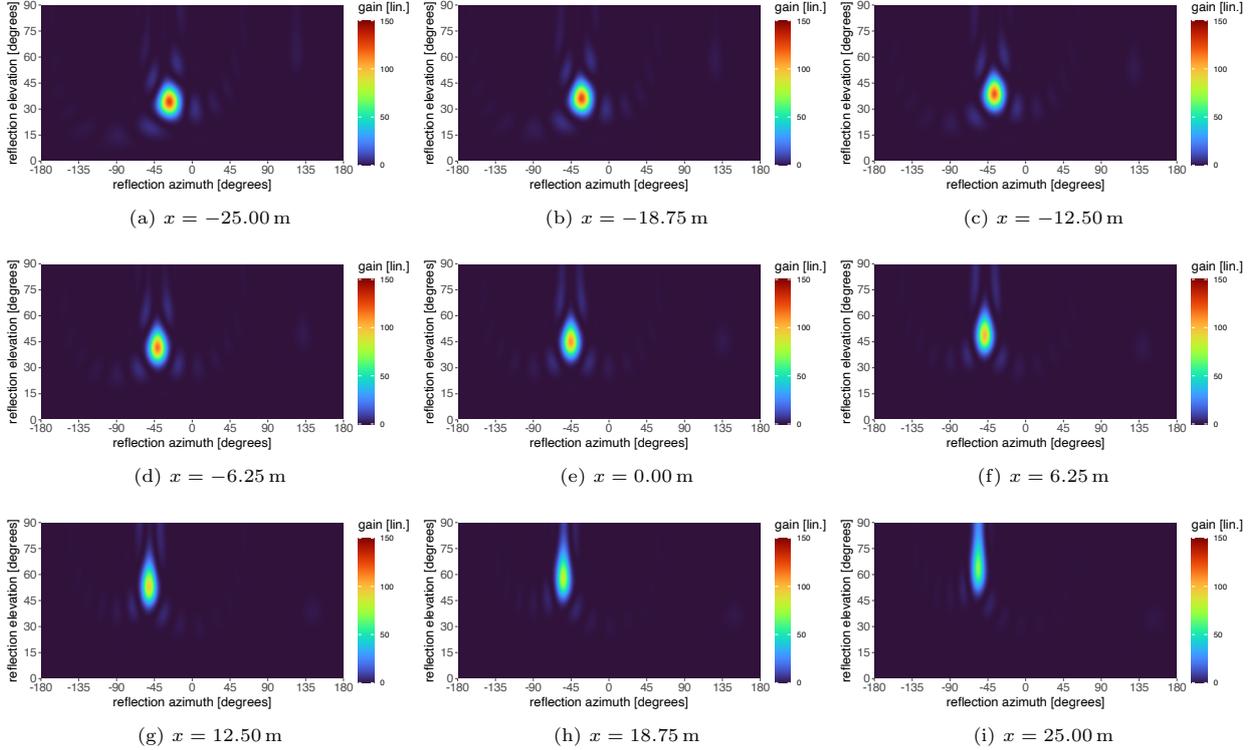


Figure 7: Far field patterns for different positions of the transmitter ($x, 100 \text{ m}, 0 \text{ m}$). The RIS is configured for the reflection ($\phi_r = -45^\circ, \theta_r = 45^\circ, \phi_i = 0^\circ, \theta_i = 0^\circ$). $p = (0, 0, 0)$, $v_1 = (1, 0, 0)$, $v_2 = (0, 0, -1)$.

in poor performance, as could be expected. In all the considered setups the gain obtained is much lower than for the counting strategy. In particular, the surface “leaks” some of the energy by partially directing it towards the normal, as it can be seen by looking at the far field pattern for the elevation angle $\theta_{r,x} = 0^\circ$. Especially in Figs. 8c and 8e, the “leaked” energy is more than the energy in the intended directions.

The counting strategy is instead more effective. For just two beams (Fig. 8b) the strategy splits the energy towards the two intended directions quite evenly, resulting in roughly 45 % and 41 % of the gain for the single beam case. For three beams (Figs. 8d and 8f) the results have a little leak towards $\phi_r = -135^\circ$, but they can still direct 24 %, 38 %, and 20 % (Fig. 8d) and 23 %, 38 %, and 23 % (Fig. 8f) of the energy in the intended directions. The reason for considering the slightly different setup in Fig. 8f is to avoid that the reflection angles differing by 90° in Fig. 8d bias the results.

In addition to the results presented in this section, we will show the impact of beam splitting in the

simulator, in particular in the no-tracking scenario (Sect. 7.5).

7.4. Perfect tracking scenario

This section describes the outcomes of the scenario featuring perfect tracking. The aim of this section is twofold: performing an initial feasibility analysis for RIS-enabled V2V communications, as well as showcasing the potential of CooperIS as a tool for the performance evaluation of next-generation network technologies for cooperative driving. We begin by examining the path loss, followed by an evaluation of the gain the RIS can provide. Path loss results can clearly be obtained analytically but we present our analysis based on CooperIS simulation, to show they match the model in [28]. In addition, results related to path loss remain valid irrespective of whether tracking is in place or not: the impact of tracking is observed in the RIS gain.

We start our analysis by describing the results for the product of distances in both T and Z scenarios (Figs. 9a and 9c). These plots illustrate the total path loss from the transmitter to the receiver as a

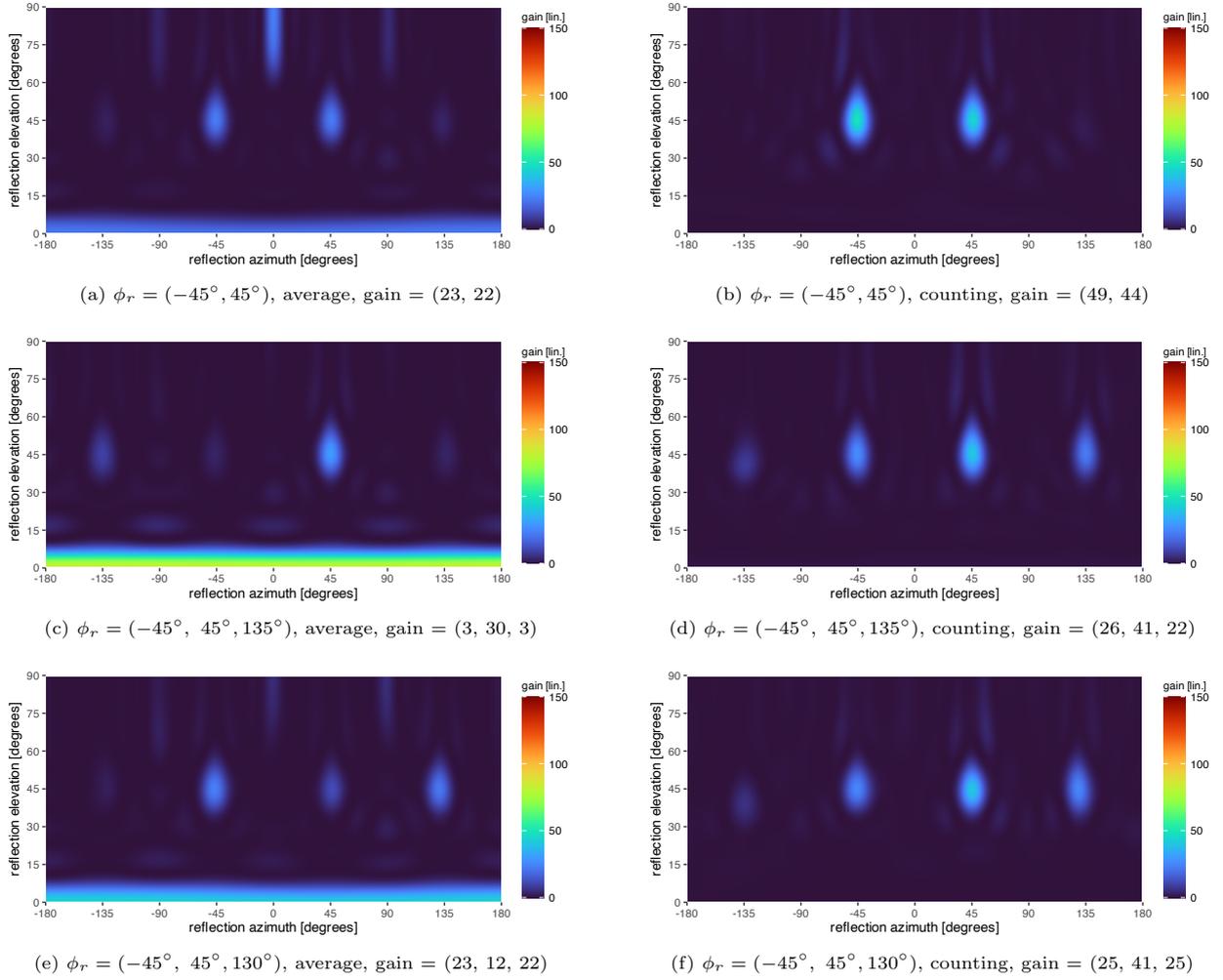


Figure 8: Far field patterns after configuring the RIS for beam splitting. The RIS is configured for the reflection ($\theta_r = 45^\circ$, $\phi_i = 0^\circ$, $\theta_i = 0^\circ$). Reported gains are in linear scale. For comparison, the gain for the single beam case is 108.

function of the length of the last hop of the path (d_2 for the T scenario, d_3 for the Z scenario) across various distances of the transmitter from the intersection. The legend reports only the two extreme values of distance (0 and 200 m). The curves in between refer to the transmitter distances listed in Tab. 2. Path loss is denoted as a negative value in dB. As it could be foreseen, the outcomes for both scenarios exhibit qualitative similarities, differing simply due to the additional path loss introduced by the longer path in the Z scenario. In quantitative terms, however, there is a significant difference. For a single reflection, the path loss spans from a minimum of 170 dB to a maximum of 220 dB, while for a double reflection, it ranges between 260 dB and

310 dB. As we are considering the product of distances model, taking Eq. (7) at a reference distance of 1 m, we obtain

$$(N_{RIS} + 1) \cdot 20 \log_{10} \left(\frac{4\pi f}{c} \right) [\text{dB}]. \quad (14)$$

At 25 GHz, each reflection causes at least 60 dB of loss. As a consequence, for single and double reflections, the path loss is going to be at least 120 dB and 180 dB, respectively.

With respect to the sum of distances (Figs. 9b and 9d), the loss is less extreme simply because the whole path is considered as a single hop. The impact of the last hop is less pronounced in this case, as the majority of the loss occurs over the initial hop.

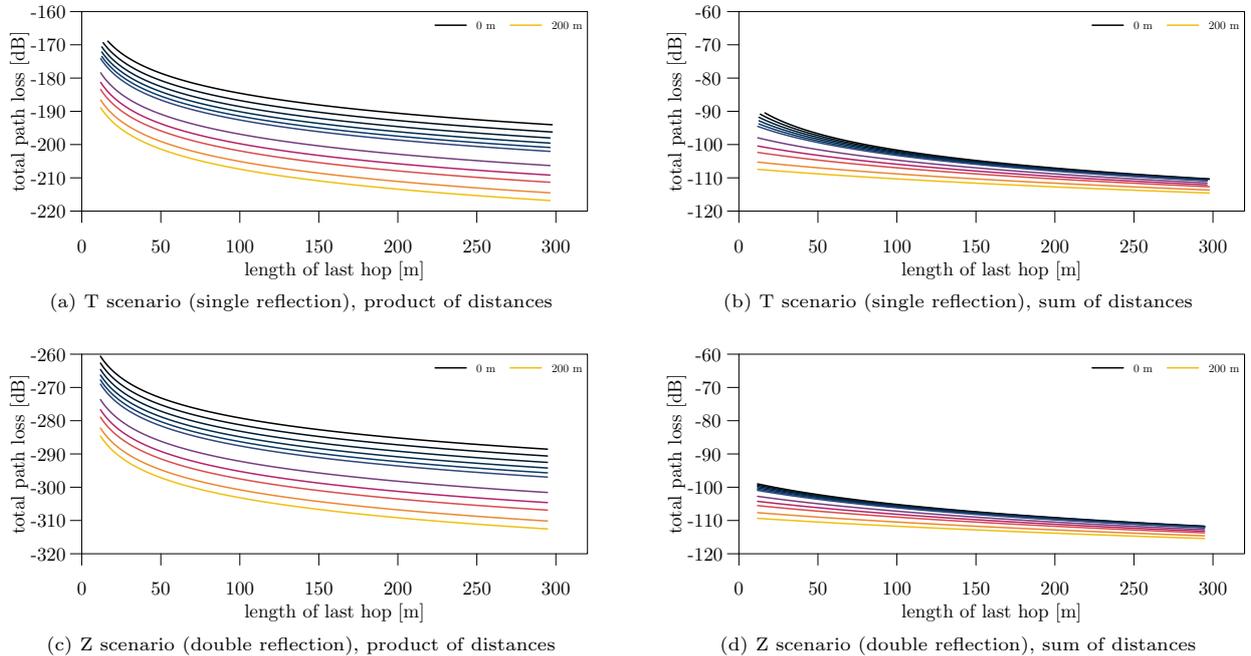


Figure 9: Path loss as a function of the last path length (without RIS and antenna gains) measured for single and double reflections, using the sum and the product of distances in the model. Different curves represent the path loss for different distances of the transmitting vehicle from the intersection.

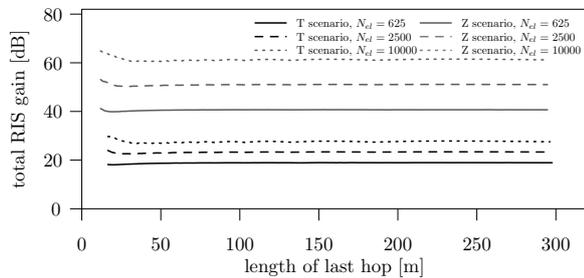


Figure 10: Total RIS gain (over whole path) measured for single and double reflections (T and Z scenarios, respectively), for different number of elements in the RIS when considering perfect tracking.

Path loss results alone are not sufficient to understand whether V2V mmWave communications through RIS are feasible or not, because such values need to be put into context. To do so, we first consider the noise floor, which directly contributes to the signal to noise ratio (SNR). In turn, noise depends on the bandwidth. For instance, a 10 MHz channel yields a noise floor of approximately -95 dBm, and for the typical 400 MHz of a mmWave channel, the noise floor can be as high as -80 dBm. Factoring in transmission power and antenna gains

(RIS included), we can translate a path loss into a base SNR. For a transmission power of 30 dBm and a noise floor of -80 dBm, the path losses in Fig. 9 can be converted into an SNR by adding 110 dB.

For the "sum of distances" path loss model, this would already result in an SNR larger than 0 dB, without considering antenna and RIS gains. In the product of distances model, the base SNR ranges between -60 dB and -110 dB for the T scenario and -150 dB and -200 dB for the Z scenario.

To further enhance the SNR, we would need to consider directional antennas. A dipole antenna can yield 9 dBi at 2.4 GHz. By using the relationship between gain and antenna effective aperture A_e [37],

$$G = \frac{4\pi f^2 A_e}{c^2} \quad (15)$$

the same antenna at 25 GHz would provide roughly 29 dBi of gain. Antennas that are more directive than dipoles could further improve performance.

The crucial point is the extent of gain the RIS can provide. We partly addressed this in Sect. 7.1 but such results are configuration-specific, so we now look into RIS gain with perfect tracking. Fig. 10 shows the total RIS gain as a function of the last hop

length for the T and Z scenarios and for different number of reflective elements. Perfect tracking results in negligible gain variations, regardless of the position of the receiving vehicle. However, the gain for a passive metasurface model and the considered number of elements (around 20 dB to 30 dB per RIS) may be insufficient, especially with multiple reflections. In such cases, it might be necessary to consider larger scale RIS deployments, active surfaces, or models like the one in [36] for further improvements.

7.5. No tracking scenario

As a final evaluation step we consider a static configuration of the RIS for the Z scenario. Here, the RIS does not track the receiver, but it focuses the beam towards a specific position of the road. We first consider a single beam pointing 50 m before the intersection and then a beam splitting configuration pointing 50 m and 200 m before the intersection. Different from previous results, we consider more fine-grained angular resolutions, in particular by setting both d_ϕ and d_θ to 0.1° .

Fig. 11 shows the gain of the second RIS in the path and the received power as a function of the length of the last hop, for different numbers of elements, and for the single and beam-splitting configurations. With respect to the gain, the graph permits us to draw different conclusions. First, Fig. 11a shows that increasing the number of elements increases the maximum gain and decreases the coverage area, as shown around 50 m, as expected. Second, as we observed in [14], the RIS can provide a positive gain even several hundred meters away from the focal point, at least for a small number of elements. When getting closer to the intersection, as the reflection azimuth and elevation angles change abruptly, the receiver experiences a sharp drop in gain.

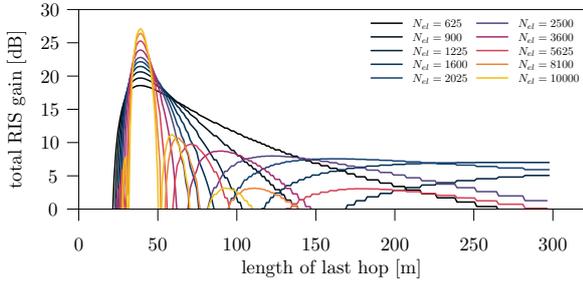
In the portion of the road before the focal point, instead we can see that increasing the number of elements does not lead to a monotonically increasing gain. To interpret this result, we have to consider the far field patterns in Fig. 5. These patterns show that as we increase the number of reflecting elements we increase the number of side lobes and nulls, together with their density. This induces the rapidly changing gain we observe in the graphs, suggesting that when the system does not actively track a node, determining the best RIS configuration is not obvious. In particular, increasing the number of elements is not necessarily the best solution. **Observing**

the curves in Fig. 11 against those in Fig. 9c (top three grey curves) for 625, 2500 and 10,000 elements, the benefit of tracking becomes apparent. In the absence of tracking, the moving vehicle eventually falls in one of the gain notches due to the fixed configuration of the RISs. The gain instability is also more pronounced as the number of RIS elements increases, as the surface’s radiation pattern involves sharper and higher-gain beams. Besides showing the benefit of tracking for vehicular scenarios, this comparison stresses a key tradeoff between the number of RIS elements and scenario coverage: the larger the number of elements, the higher the maximum gain, the more often the tracking algorithm should re-align the reflected beam to the actual location of the user. Conversely, the lower the number of elements, the lower the maximum gain, the more graceful the gain degradation when the receiver is off the main reflection direction.

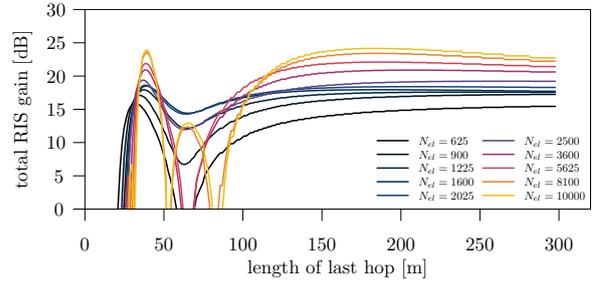
We can observe the impact from the perspective of the vehicle located on the vertical road of the Z scenario as well. Fig. 12 plots the received power as function of the number of reflective elements, for different transmitter positions. As for Fig. 9, the legend shows the two extreme values only but the curves refer to the distances listed in Tab. 2. Intuitively, we might expect that increasing the number of elements would simply make the beam narrower and consequently reduce the received power, because the first RIS is focusing more and more the beam towards the second one. However the received power fluctuates several times before decreasing monotonically.

Finally, in Fig. 11b we show how the coding combiner algorithm of Sect. 5 affects the gain when used in a beam-splitting configuration. Overall, the maximum gain is reduced as the energy is directed towards a larger area. As in the previous case, the behavior of the gain is far from being trivial. For the highest number of reflecting elements, the gain around the intended focus zones is the highest. Yet, the 50 m beam covers a very narrow portion of the street, whereas the 200 m beam produces the highest gain between roughly 125 m and 300 m. Between 50 m and 100 m the gain falls towards two nulls, but between them it provides the third best gain. Looking at other configurations in the same 50 m-100 m region, for 625 elements we find a null around 65 m. For 1600 and 2025 elements we just see a little loss of gain in that region but no “deep” null.

The results in this section show that providing the optimal V2V connectivity through RIS is far



(a) single beam (50 m into the intersection)



(b) beam splitting (50 and 200 m into the intersection)

Figure 11: Gain of the second RIS for the Z scenario (double reflection) with no tracking and product of distances, a different number of elements in the RIS, and single and beam-splitting configurations.

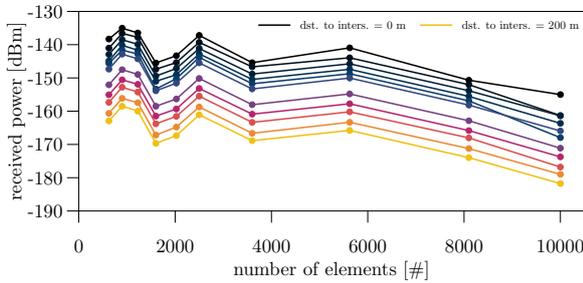


Figure 12: Received power for the Z scenario (double reflection) with no tracking and product of distances as function of the number of elements in the RIS, for different distances of the transmitter from the intersection. The received power refers to the vehicle receiving packet on the second path.

from trivial and that it presents several research challenges. At the same time, they showcase the potential of CooperRIS for the performance evaluation of such systems and that it can become a fundamental tool for the development of this technology.

8. Conclusions, discussion, and future work

The motivations of this work were multiple. First, after identifying the interest of the research community in RIS-enabled mmWave communication for future cooperative driving applications, we discovered that existing studies base their analysis mostly on MATLAB-based numerical simulations. This provides a limited playground, as it remains difficult to model and simulate realistic cooperative driving applications. Second, there exist mature and popular DES frameworks such as PLEXE and Veins, that however still lack mmWave communication capabilities, let alone RISs. To bridge the gap, this paper develops and releases CooperRIS, a module

for the Veins ecosystem enabling the possibility of investigating such systems.

Through CooperRIS, we have performed a feasibility study, showing the challenges that RIS-based V2V communications pose both in terms of path loss and proper metasurface configuration. Moreover, we proposed an algorithm for combining multiple RIS configurations into a single one, again using CooperRIS as a performance evaluation tool. The results prove the effectiveness of CooperRIS for the development and the performance evaluation of future cooperative driving communication systems.

While this work improves the state of the art, there is still a long road ahead. The models in CooperRIS require heavy mathematical computations, and the code it is not yet optimized to exploit parallel computing architectures. In addition, the mmWave communication model is simple, and more realistic ones need to be employed. From a research perspective, there is the need to understand how to properly configure a RIS to achieve the best performance, including physical characteristics such as the proper number of elements. RIS-based V2V communication is especially challenging because, different from cellular networks, we have two nodes to track rather than a single user equipment. Finally, we will need to understand how to schedule resources among multiple concurring pairs of vehicles that wish to exploit a RIS at the same time. This might include using the coding combination algorithm we propose in this paper, perform time multiplexing, or even using configuring sub-areas of the RIS for different users. We believe this work paves the road towards new research opportunities in this direction.

Acknowledgements

This research has received support from the Italian PRIN program, prot. no. 2022SKLZAY, funded by the European Union under NextGenerationEU and the project BRIDGE2HORIZON/0823A/004 (AUTOMETA) which is implemented under the Cohesion Policy Funds "THALEIA 2021-2027" with EU co-funding. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the EU or the EU REA. Neither the EU nor the granting authority can be held responsible for them.

References

- [1] V. Lesch, M. Breitbach, M. Segata, C. Becker, S. Kounev, C. Krupitzer, An Overview on Approaches for Coordination of Platoons, *IEEE Transactions on Intelligent Transportation Systems* 23 (8) (2022) 10049–10065. doi:10.1109/TITS.2021.3115908.
- [2] H. Wang, Y. Huang, A. Khajepour, Y. Zhang, Y. Rasekhipour, D. Cao, Crash Mitigation in Motion Planning for Autonomous Vehicles, *IEEE Transactions on Intelligent Transportation Systems* 20 (9) (2019) 3313–3323. doi:10.1109/tits.2018.2873921.
- [3] F. Dressler, G. S. Panmu, F. Hagenauer, M. Gerla, T. Higuchi, O. Altintas, Virtual Edge Computing Using Vehicular Micro Clouds, in: *IEEE International Conference on Computing, Networking and Communications (ICNC 2019)*, IEEE, Honolulu, HI, 2019. doi:10.1109/ICNC.2019.8685481.
- [4] M. B. Mollah, H. Wang, M. A. Karim, H. Fang, mmWave Enabled Connected Autonomous Vehicles: A Use Case with V2V Cooperative Perception, *IEEE Network To appear* (2024). doi:10.1109/mnet.2023.3321520.
- [5] M. Noor-A-Rahim, Z. Liu, H. Lee, M. O. Khyam, J. He, D. Pesch, K. Moessner, W. Saad, H. V. Poor, 6G for Vehicle-to-Everything (V2X) Communications: Enabling Technologies, Challenges, and Opportunities, *Proceedings of the IEEE* 110 (6) (2022) 712–734. doi:10.1109/jproc.2022.3173031.
- [6] M. Xiao, S. Mumtaz, Y. Huang, L. Dai, Y. Li, M. Matthaiou, G. K. Karagiannidis, E. Björnson, K. Yang, C.-L. I, A. Ghosh, Millimeter Wave Communications for Future Mobile Networks, *IEEE Journal on Selected Areas in Communications* 35 (9) (2017) 1909–1935. doi:10.1109/JSAC.2017.2719924.
- [7] C. Liaskos, A. Tsiolaridou, S. Nie, A. Pitsillides, S. Ioannidis, I. F. Akyildiz, On the Network-Layer Modeling and Configuration of Programmable Wireless Environments, *IEEE/ACM Transactions on Networking* 27 (4) (2019) 1696–1713. doi:10.1109/tnet.2019.2925658.
- [8] D. Tyrovolas, S. A. Tegos, V. K. Papanikolaou, Y. Xiao, P.-V. Mekikis, P. D. Diamantoulakis, S. Ioannidis, C. Liaskos, G. K. Karagiannidis, Zero-Energy Reconfigurable Intelligent Surfaces (zeRIS), *IEEE Transactions on Wireless Communications* (2023). doi:10.1109/twc.2023.3336956.
- [9] S. A. Tegos, D. Tyrovolas, P. D. Diamantoulakis, C. Liaskos, G. K. Karagiannidis, On the Distribution of the Sum of Double-Nakagami- m Random Vectors and Application in Randomly Reconfigurable Surfaces, *IEEE Transactions on Vehicular Technology* 71 (7) (2022) 7297–7307. doi:10.1109/tvt.2022.3164846.
- [10] A. Ptilakis, O. Tsilipakos, F. Liu, K. M. Kossifos, A. C. Tasolamprou, D.-H. Kwon, M. S. Mirmoosa, D. Manassis, N. V. Kantartzis, C. Liaskos, M. A. Antoniadis, M. Pappageorgiou, C. M. Soukoulis, M. Kafesaki, S. Tretyakov, A Multi-Functional Reconfigurable Metasurface: Electromagnetic Design Accounting for Fabrication Aspects, *IEEE Transactions on Antennas and Propagation* 69 (3) (2021) 1440–1454. doi:10.1109/tap.2020.3016479.
- [11] R. Guirado, G. Perez-Palomino, M. Cano-Garcia, M. A. Geday, E. Carrasco, mm-Wave Metasurface Unit Cells Achieving Millisecond Response Through Polymer Network Liquid Crystals, *IEEE Access* 10 (2022) 127928–127938. doi:10.1109/access.2022.3226601.
- [12] D. Tyrovolas, S. A. Tegos, E. C. Dimitriadou-Panidou, P. D. Diamantoulakis, C. Liaskos, G. K. Karagiannidis, Performance Analysis of Cascaded Reconfigurable Intelligent Surface Networks, *IEEE Wireless Communications Letters* 11 (9) (2022) 1855–1859. doi:10.1109/lwc.2022.3184635.
- [13] M. Segata, R. Lo Cigno, T. Harde, J. Heinovski, M. Schettler, B. Bloessl, C. Sommer, F. Dressler, Multi-Technology Cooperative Driving: An Analysis Based on PLEXE, *IEEE Transactions on Mobile Computing* 22 (8) (2023) 4792–4806. doi:10.1109/TMC.2022.3154643.
- [14] M. Segata, M. Lestas, P. Casari, T. Saeed, D. Tyrovolas, G. K. Karagiannidis, C. Liaskos, Enabling Cooperative Autonomous Driving Through mmWave and Reconfigurable Intelligent Surfaces, in: *18th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2023)*, IEEE, Madonna di Campiglio, Italy, 2023. doi:10.23919/WONS57325.2023.10062109.
- [15] M. Segata, P. Casari, M. Lestas, D. Tyrovolas, T. Saeed, G. K. Karagiannidis, C. Liaskos, On the Feasibility of RIS-enabled Cooperative Driving, in: *14th IEEE Vehicular Networking Conference (VNC 2023)*, IEEE, Istanbul, Turkey, 2023. doi:10.1109/VNC57357.2023.10136324.
- [16] Y. Zhu, B. Mao, N. Kato, Intelligent Reflecting Surface in 6G Vehicular Communications: A Survey, *IEEE Open Journal of Vehicular Technology* 3 (2022) 266–277. doi:10.1109/ojvt.2022.3177253.
- [17] Y. Chen, Y. Wang, L. Jiao, Robust Transmission for Reconfigurable Intelligent Surface Aided Millimeter Wave Vehicular Communications With Statistical CSI, *IEEE Transactions on Wireless Communications* 21 (2) (2022) 928–944. doi:10.1109/twc.2021.3100492.
- [18] J. Wang, W. Zhang, X. Bao, T. Song, C. Pan, Outage Analysis for Intelligent Reflecting Surface Assisted Vehicular Communication Networks, in: *IEEE Global Communications Conference (GLOBECOM 2020)*, IEEE, Taipei, Taiwan, 2020. doi:10.1109/globecom42002.2020.9322158.
- [19] Y. Ai, F. A. P. deFigueiredo, L. Kong, M. Cheffena, S. Chatzinotas, B. Ottersten, Secure Vehicular Communications Through Reconfigurable Intelligent Surfaces, *IEEE Transactions on Vehicular Technology* 70 (7) (2021) 7272–7276. doi:10.1109/tvt.2021.3088441.
- [20] Y. U. Ozcan, O. Ozdemir, G. K. Kurt, Reconfigurable Intelligent Surfaces for the Connectivity of Autonomous Vehicles, *IEEE Transactions on Vehicular Technology* 70 (3) (2021) 2508–2513. doi:10.1109/tvt.2021.3060667.
- [21] T. Saeed, W. Aziz, A. Pitsillides, V. Vassiliou, I. F. Akyildiz, H. Taghvaei, S. Abadal, C. Liaskos, A. Tsiolaridou, S. Ioannidis, E. Emoyon-Iredia, M. Lestas,

- On the Use of Programmable Metasurfaces in Vehicular Networks, in: 22nd IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2021), IEEE, Virtual Conference, 2021. doi:10.1109/spawc51858.2021.9593176.
- [22] W. Lv, J. Bai, Q. Yan, H. M. Wang, RIS-Assisted Green Secure Communications: Active RIS or Passive RIS?, IEEE Wireless Communications Letters 12 (2) (2023) 237–241. doi:10.1109/lwc.2022.3221609.
- [23] H. Niu, Z. Lin, K. An, X. Liang, Y. Hu, D. Li, G. Zheng, Active RIS-Assisted Secure Transmission for Cognitive Satellite Terrestrial Networks, IEEE Transactions on Vehicular Technology 72 (2) (2023) 2609–2614. doi:10.1109/tvt.2022.3208268.
- [24] E. Basar, I. Yildirim, Reconfigurable Intelligent Surfaces for Future Wireless Networks: A Channel Modeling Perspective, IEEE Wireless Communications 28 (3) (2021) 108–114. doi:10.1109/mwc.001.2000338.
- [25] A. Habib, I. Khaled, A. El Falou, C. Langlais, Extended NYUSIM-based MmWave Channel Model and Simulator for RIS-Assisted Systems, in: IEEE European Conference on Networks and Communications (EuCNC 2023 & 6G Summit), IEEE, Gothenburg, Sweden, 2023. doi:10.1109/eucnc/6gsummit58263.2023.10188290.
- [26] S. Ju, O. Kanhere, Y. Xing, T. S. Rappaport, A Millimeter-Wave Channel Simulator NYUSIM with Spatial Consistency and Human Blockage, in: IEEE Global Communications Conference (GLOBECOM 2019), IEEE, Waikoloa, HI, 2019. doi:10.1109/globecom38437.2019.9013273.
- [27] Q. Gu, D. Wu, X. Su, H. Wang, J. Cui, Y. Yuan, System-level Simulation of RIS assisted Wireless Communications System, in: IEEE Global Communications Conference (GLOBECOM 2022), IEEE, Rio de Janeiro, Brazil, 2022. doi:10.1109/globecom48099.2022.10000700.
- [28] H. Taghvaei, S. Abadal, A. Ptilakis, O. Tsilipakos, A. C. Tasolamprou, C. Liaskos, M. Kafesaki, N. V. Kantartzis, A. Cabellos-Aparicio, E. Alarcon, Scalability Analysis of Programmable Metasurfaces for Beam Steering, IEEE Access 8 (2020) 105320–105334. doi:10.1109/access.2020.3000424.
- [29] W. Tang, X. Chen, M. Z. Chen, J. Y. Dai, Y. Han, M. Di Renzo, S. Jin, Q. Cheng, T. J. Cui, Path Loss Modeling and Measurements for Reconfigurable Intelligent Surfaces in the Millimeter-Wave Frequency Band, IEEE Transactions on Communications 70 (9) (2022) 6259–6276. doi:10.1109/tcomm.2022.3193400.
- [30] G. Nardini, D. Sabella, G. Stea, P. Thakkar, A. Virdis, Simu5G – An OMNeT++ Library for End-to-End Performance Evaluation of 5G Networks, IEEE Access 8 (2020) 181176–181191. doi:10.1109/access.2020.3028550.
- [31] C. Sommer, R. German, F. Dressler, Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis, IEEE Transactions on Mobile Computing 10 (1) (2011) 3–15. doi:10.1109/TMC.2010.133.
- [32] P. Alvarez Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, E. Wießner, Microscopic Traffic Simulation using SUMO, in: 21st IEEE International Conference on Intelligent Transportation Systems (ITSC 2018), IEEE, Maui, HI, 2018, pp. 2575–2582. doi:10.1109/ITSC.2018.8569938.
- [33] M. Drago, T. Zugno, M. Polese, M. Giordani, M. Zorzi, MilliCar: An Ns-3 Module for MmWave NR V2X Networks, in: Workshop on ns-3 (WNS3 2020), ACM, 2020. doi:10.1145/3389400.3389402.
- [34] W. Tang, M. Z. Chen, X. Chen, J. Y. Dai, Y. Han, M. Di Renzo, Y. Zeng, S. Jin, Q. Cheng, T. J. Cui, Wireless Communications With Reconfigurable Intelligent Surface: Path Loss Modeling and Experimental Measurement, IEEE Transactions on Wireless Communications 20 (1) (2021) 421–439. doi:10.1109/twc.2020.3024887.
- [35] C. Sommer, D. Eckhoff, R. German, F. Dressler, A Computationally Inexpensive Empirical Model of IEEE 802.11p Radio Shadowing in Urban Environments, in: 8th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2011), IEEE, Bardonecchia, Italy, 2011, pp. 84–90. doi:10.1109/WONS.2011.5720204.
- [36] D. Tyrovolas, S. A. Tegos, P. D. Diamantoulakis, G. K. Karagiannidis, Synergetic UAV-RIS Communication With Highly Directional Transmission, IEEE Wireless Communications Letters 11 (3) (2022) 583–587. doi:10.1109/lwc.2021.3136912.
- [37] T. S. Rappaport, Wireless Communications: Principles and Practice, 2nd Edition, Prentice Hall, Upper Saddle River, NJ, 2001.

Appendix A. Source code structure

To ease the customization of CooperRIS by other researchers, this appendix describes how the source files are organized, together with a description of their functionalities. The following tree shows the list of the main source files included within CooperRIS. In here we omit other general files such as Makefiles and we only focus on the source code to understand where different parts are implemented.

```

cooperis
|-- src
    |-- cooperis
        |-- CarRis.ned
        |-- DeciderResultRis.h
        |-- DeciderRis.cc
        |-- DeciderRis.h
        |-- NicRis.ned
        |-- PhyLayerRis.cc
        |-- PhyLayerRis.h
        |-- PhyLayerRis.ned
        |-- Ris.ned
        |-- SUMOBaseMobility.cc
        |-- SUMOBaseMobility.h
        |-- SUMOBaseMobility.ned
        |-- analogueModel
        | |-- RisPathLoss.cc
        | |-- RisPathLoss.h
        | |-- SimpleObstacleShadowingRis.cc
        | |-- SimpleObstacleShadowingRis.h
    |-- mac
        |-- MacLayerRis.cc
        |-- MacLayerRis.h
        |-- MacLayerRis.ned
    |-- messages
        |-- AirFrameRis.msg
    |-- utility
        |-- ReconfigurableIntelligentSurface.cc
        |-- ReconfigurableIntelligentSurface.h
        |-- Utils.cc
        |-- Utils.h

```

To describe the role of each source file, we follow the same order of the above list.

CarRis.ned This file defines a network node for a vehicle, i.e., its network stack and its applications.

DeciderRis.*, **DeciderResultRis.h** These files takes care of computing the probability of reception of frames depending on signal to interference plus noise ratio (SINR) and MCS, and deciding whether the frame was correctly received or not. Such decision is then stored within a DeciderResult and returned to the PHY layer.

NicRis.ned This file defines a NIC for a RIS-enabled communication node. This can be used both by network nodes and RIS. For standard network nodes, both the MAC and PHY layers are used. For a RIS, instead, only the PHY layer.

PhyLayerRis.* Definition and implementation of the PHY layer. This is where the Listing 1 is actually implemented.

Ris.ned Definition of a network node for a RIS, basically including a NicRis and a mobility module used to statically place the RIS in the simulation.

SUMOBaseMobility.* Mobility module used to ease the placement of a RIS in a simulation. Using this module it is possible to specify the position of a RIS using SUMO coordinates, for example at an intersection. This module takes care of converting SUMO coordinates into OMNeT++ coordinates automatically.

RisPathLoss.* This module takes care of computing the path loss depending on the chosen model (sum or product of distances) as well as applying the RIS gain to an incoming signal.

SimpleObstacleShadowingRis.* This module extends the obstacle shadowing model of Veins by tagging with a flags frames that were crossing a building.

SimplePathlossModelRis.* This module extends the obstacle shadowing model of Veins by tagging with a flags frames that were crossing a building.

MacLayerRis.* Implementation of the MAC layer for a RIS-enabled communication node.

AirFrameRis.msg High-level representation of a frame, which includes the metadata defined in Sect. 4.2, such as incidence angles, the list of RISs which reflected the signal, etc.

ReconfigurableIntelligentSurface.* Actual implementation of the RIS model in Sect. 3. This includes RIS coding and gain computation. This class does not depend on OMNeT++ and can be used as a standalone library in other software projects.

Utils.* Implementation of geometric utilities, such as conversions from Cartesian to spherical coordinates.