2025-August-21 Tech. Report 000

Philip Rosberg[1]
Josef Holmner[2]

AGX Sensor

Theoretical Background and Method

Summary:

Simulating sensor data with physical fidelity is increasingly vital in the design, testing, and validation of *Autonomous System*s. The *AGX Sensor* module of the *AGX Dynamics* simulation suite, offers a unified platform for real-time virtual sensing, enabling the generation of synthetic sensor outputs across a wide array of modalities. By operating within a dedicated *Sensor Environment* mostly decoupled from the main dynamics simulation, this module allows for flexible integration of sensor simulation while allowing the main dynamics simulation to be run by itself. This module supports configurable sensor types, such as *Lidar*s, *Inertial Measurement Unit*s (*IMU*s) and *Accelerometer*s among others, each designed to replicate realistic sensing conditions, environmental interactions, and common sources of measurement uncertainty. This capability facilitates the development of robust control strategies, perception algorithms, and autonomy pipelines, making the *AGX Sensor* module a versatile tool for research and development in robotics, simulation-based testing, and virtual prototyping.

Algoryx Simulation
algoryx.com
(+46) 90 348 49 90
contact@algoryx.com

[1]philip.rosberg@algoryx.com
[2]josef.holmner@algoryx.com

Contents

1 Introduction

AGX Sensor is the sensor simulation module of the *AGX Dynamics* simulation software. It provides a variety of sensors capable of performing real-time sensing in tandem with the rest of the *AGX Dynamics* simulation suite. Each sensor is easily configured and provides its output signal in a common standardized way.

The *AGX Sensor* module is designed with the intent of providing a way to conveniently model physically accurate sensors and integrate them into the overarching simulation. These sensors can then be used to produce synthetic sensor data to verify *Control System* functionality and develop systems for *Machine Autonomy*.

1.1 Environment

All sensors in the *AGX Sensor* module reside within a dedicated *Sensor Environment*. This *Sensor Environment* is a parallel to the main dynamics simulation of *AGX Dynamics*, and stores both the sensors themselves and information about how the sensors should interpret the world. The separation between the *Sensor Environment* and the main dynamics simulation allows for a one-way dependence of the *Sensor Environment* on the main dynamics simulation while allowing the dynamics simulation to be run without sensor support if needed. Which information is relevant to add to the *Sensor Environment* depends on which sensors are used, and which effects are desired to be captured in the sensor data.

2 Lidar

Light Detection and Ranging (*Lidar*) is a kind of sensor that is predominantly used to measure distances and map the space around the sensor [32, 43, 48]. These *Lidar* sensors can come in multiple different configurations depending on their intended use, but their main working principle is that they emit focused beams of light in the near infrared region and measures distance using the returning signal.

The *AGX Sensor* module provides a *Lidar* sensor implementation based on multi-step raytracing. This implementation is thus suitable to be used to model nearly all *Lidar* sensors that use pulsed unpolarized light at a single or a narrow wavelength band, such as nearly all vehicle *Lidar* sensors [24, 35, 46]. In the case of generating synthetic data for *Control System* verification or *Machine Autonomy*, this kind of *Lidar* implementation covers effectively all *Lidar* sensors commonly used for the real world analogue.

2.1 Raytrace Process and Return Selection

The *Lidar* implementation in the *AGX Sensor* module generates its returns through a process consisting of a number of ordered steps, starting with ray creation, proceeding to ray manipulation, raytracing, point assembly and ending with post-processing of the point cloud. Through the ray creation and ray manipulation steps, the *Lidar* scan pattern and scan slice being processed that step are configured. These ray configurations are then passed on as initial conditions to the raytrace step. Any solid objects registered for raytracing visibility in the *Sensor Environment* are also made available to the raytrace step, together with their respective *Surface Material*s.

During raytracing, surface geometric parameters and material properties are registered at each interface, or ray hit, in a binary heap, a process which can be seen visualized in Figure 1. At an interface a single incident ray is split into one reflection ray and one transmission ray, consuming each their own sub-branch of the heap following the interface node. The parameters stored are those necessary to later compute returns in the point assembly step of the raytrace process, such as reflection and transmission intensities and the parameters required to generate *Atmospheric Return*s.

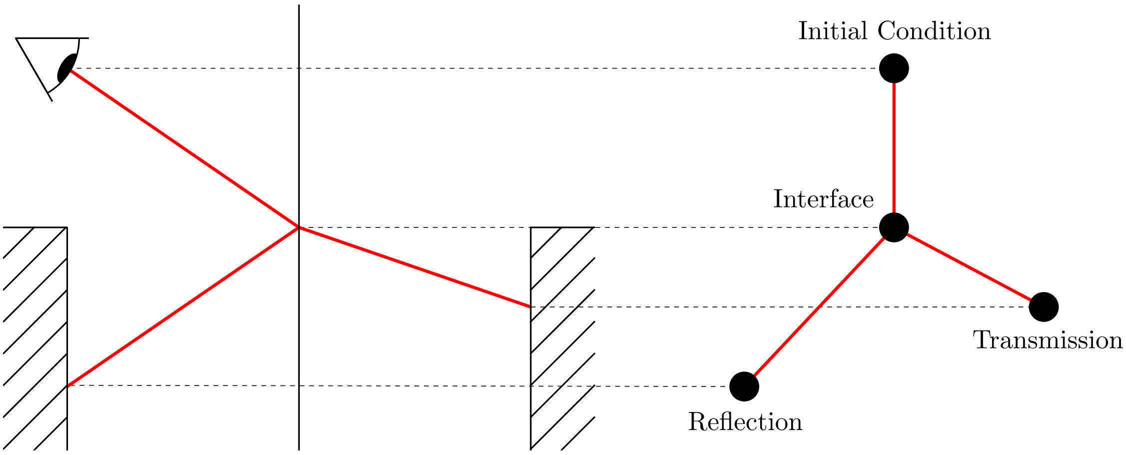

Figure 1: Illustration of the raytrace step. As the ray traverses the *Sensor Environment* (as illustrated to the left), the geometric parameters and material properties are stored in a binary heap at each interface, or ray hit (as illustrated to the right).

On assembly of the *Lidar* raytrace returns, the binary heap as visualized in Figure 1 is traversed. At interface nodes belonging to *Surface Material*s, a return is generated at the ray-interface intersection's *Optical Path Length* from the ray emission source. Conversely, if the interface node has parameters to generate *Atmospheric Return*s, a return is created at a displaced *Optical Path Length* behind the interface. Each return is attenuated by its preceding *Optical Path Length* between interfaces and the reflection or transmission factors at each preceding interface. As the final part of the point assembly, the return of with the strongest intensity is selected, and the ray emission direction is scaled by this selected return's *Optical Path Length* to give a resulting *Lidar* return point.

Finally, the point cloud generated by the point assembly step is optionally post-processed. This step is used to add additional sources of error, like *Detection Distance Inaccuracy*, or to transform the points from the world frame to the local *Lidar* frame.

2.2 Surface Material

Solid objects in the *Sensor Environment* have opaque surfaces. Each one of the surfaces of these objects has its interactions with the *Lidar* laser light described using a *Surface Material*. This *Surface Material* describes the reflection and absorption characteristics when the *Lidar* laser light interacts with the surface.

In the general infinitesimal case, the reflection and absorption characteristics of a surface, for light incident in direction l and reflected in the view direction v, may be described using the *Reflectance Equation* [4, 25, 39],

$$L(v) = \int_{\Omega} L_i(l) f(l, v)(l \cdot n) \, \mathrm{d}\omega \, . \tag{1}$$

The *Reflectance Equation* in Equation (1) provides a convenient way to generally model the outgoing light radiance L from an infinitesimal surface point, by integrating the incident light radiance L_i over the surface normal hemisphere Ω, as illustrated in Figure 2. In turn, this surface point has its reflection characteristics described by the *Bidirectional Reflectance Distribution Function* (*BRDF*), f in Equation (1).

However, for the generally small *Lidar* beam divergence angle, θ_D, and detection cone angle, the radiance solid angle of propagation can be approximated using a cone

$$L \approx \pi\theta_D^2 I, \tag{2}$$

which provides a relation to the *Lidar* laser beam intensity I [39]. Applying this approximation of Equation (2) to both the outgoing and incident light radiance in *Reflectance Equation* in Equation (1), allows the *Reflectance Equation* to be discretized [39] into

$$I(v) = \pi \sum_k I_k f(l_k, v)(l_k \cdot n), \tag{3}$$

by evaluating the integral over the surface normal hemisphere of Figure 2.

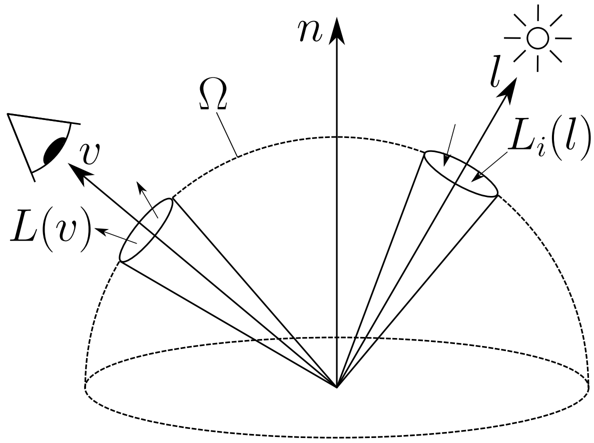

Figure 2: Configuration of the *Reflectance Equation* of Equation (1) [4, 25, 39].

It is this discretized *Reflectance Equation* in Equation (3) which the *Lidar* implementation in the *AGX Sensor* module uses for all opaque surfaces to determine the light intensity reflected along the specular reflection path and back towards the incidence direction. All the light intensity which is not reflected along one of these directions is assumed to either be absorbed in the surface or be weak enough to not affect the resulting *Lidar* return selection if diffusely reflected in another direction.

Thus, for each *Surface Material* it is the *BRDF*, f in Equation (3), which is parameterized to describe the light interaction mechanics for that specific *Surface Material*.

2.2.1 Lambertian Diffuse Surface

The *Lidar* implementation in the *AGX Sensor* module provides its most basic *Surface Material* through the opaque and perfectly smooth diffuse *Lambertian Reflector*. This material has its *BRDF* [4, 25] defined as

$$f(l, v) = \frac{C_{\text{Diffuse}}}{\pi}, \tag{4}$$

which is found from *Lambert's Cosine Law* [29] relating the light incidence vector l and the view direction v, while taking into account the intensity scaling resulting from the reflectance equation in Equation (1) or Equation (3). The *BRDF* for the *Lambertian Reflector* in Equation (4) requires one single parameter to be specified, for the diffuse reflectance C_{Diffuse}.

2.2.2 Two-layer GGX and Oren-Nayar Surface

For more complex surfaces where the *Lambertian Reflector* may not be entirely adequate the *Lidar* implementation in the *AGX Sensor* module additionally provides an opaque two-layer *Surface Material*. This material combines a microfacet specular top-layer with a rough diffuse secondary layer to achieve coverage of mirror-like surfaces, rough surfaces with a shiny coat and a wide range of non-smooth diffuse surfaces.

To achieve the most control and physical accuracy as possible for the light interactions with the *Surface Material* the full set of complex *Fresnel Equations* is used. The power ratios in the different polarization directions for the *Fresnel Equations* are defined as

$$\begin{aligned} r_s &= \frac{\tilde{n}_1 \cos\theta_1 - \tilde{n}_2 \cos\theta_2}{\tilde{n}_1 \cos\theta_1 + \tilde{n}_2 \cos\theta_2}, \qquad & r_p &= \frac{\tilde{n}_1 \cos\theta_2 - \tilde{n}_2 \cos\theta_1}{\tilde{n}_1 \cos\theta_2 + \tilde{n}_2 \cos\theta_1}, \\ t_s &= \frac{2\tilde{n}_1 \cos\theta_1}{\tilde{n}_1 \cos\theta_1 + \tilde{n}_2 \cos\theta_2}, \qquad & t_p &= \frac{2\tilde{n}_1 \cos\theta_1}{\tilde{n}_1 \cos\theta_2 + \tilde{n}_2 \cos\theta_1}, \end{aligned} \tag{5}$$

given the angle θ_1 between the incident light direction l and the surface normal, the angle θ_2 between the refracted light direction and the negative surface normal, and given the complex refractive indices of the incidence medium, $\tilde{n}_1$, and the refraction medium, $\tilde{n}_2$ [20, 36, 39]. As the laser used by the *Lidar* implementation in the *AGX Sensor* module is assumed to be unpolarized, the total reflected and transmitted power ratio can be found by averaging the power ratios from each of the polarizations [36, 39] in Equation (5) as

$$R(l,n) = \tfrac{1}{2}\left(|r_s|^2 + |r_p|^2\right), \qquad T(l,n) = \tfrac{\tilde{n}_2\theta_2}{2\tilde{n}_1\theta_1}\left(|t_s|^2 + |t_p|^2\right), \tag{6}$$

for a laser incident in direction l, from a medium of complex refractive index $\tilde{n}_1$, onto a surface with normal n to medium of complex refractive index $\tilde{n}_2$.

The microfacet specular top-layer is created starting from the base microfacet *BRDF* [1, 4, 25, 28, 39] of

$$f_{\mathrm{Top}}(l,v) = \frac{R(l,h)G(l,v,h)D(h)}{4(l\cdot n)(v\cdot n)}, \tag{7}$$

which assumes a surface consisting of more or less randomly aligned specular microfacets, where the reflected intensity off of every microfacet is calculated using the unpolarized power ratio from the *Fresnel Equations* of Equation (6), using the half-vector,

$$h = \frac{1}{2}(v + l),$$

as the surface normal, to limit reflection to only those microfacets that are visible from the view direction v [4, 25].

$D(h)$ in Equation (7) gives the distribution of the amount of microfacets capable of reflecting specularly, and is for this *Surface Material* specified using the *GG-X* distribution [1, 4, 28, 39], defined as

$$D(h) = \frac{2\sigma^2_{\mathrm{Beckman}}}{\pi\left((2\sigma^2_{\mathrm{Beckman}})(h\cdot n)^2 + 1\right)^2},$$

where $\sigma_{\mathrm{Beckman}}$ is the *Beckman Roughness*, defined as the root-mean-square of the slopes of the surface microfacets.

The $G(l,v,h)$ function in the specular top-layer microfacet *BRDF* of Equation (7) defines the geometric attenuation of the reflected light due to the bumpy nature of the microfacet surface [4, 25]. This function can usually be separated into the effects of masking and shadowing [4, 23] as

$$G(l, v, h) = G_{\text{Mask}}(l, h) G_{\text{Shadow}}(v, h). \tag{8}$$

For this *Surface Material*, both the masking and shadowing are specified using *Smith's G Function* [4, 39] defined by

$$\begin{aligned} G_{\text{Smith}}(l, h) &= \frac{1}{1 + \Lambda(a)}, \\ a &= \frac{(l \cdot h)}{\sqrt{2\sigma_{\text{Beckman}}(1 - (l \cdot h)^2)}}, \end{aligned} \tag{9}$$

where the Λ function becomes

$$\Lambda_{\text{GG-X}}(a) = \frac{-1 + \sqrt{1 + \frac{1}{a^2}}}{2},$$

when the *GG-X* distribution is used in Equation (7). However, when using *Smith's G Function* from Equation (9) for both the masking and shadowing effects the geometric attenuation function must be assembled as

$$G(l, v, h) = \frac{1}{1 + G_{\text{Smith}}(l, h) + G_{\text{Smith}}(v, h)}, \tag{10}$$

to compensate for height correlation between the microfacets [4, 39].

The secondary rough diffuse layer uses the microfacet diffuse *Oren-Nayar Model* with an improvement suggested by Fujii in "A tiny improvement of Oren-Nayar reflectance model" [16, 34, 39] formulated as

$$f_{\text{Diffuse}}(l, v) = \frac{C_{\text{Diffuse}}}{\pi} (l \cdot n) \left(A + B \frac{s}{t} \right), \tag{11}$$

with the auxiliary components

$$\begin{aligned} A &= 1 - 0.5 \frac{\sigma^2_{\text{Oren-Nayar}}}{\sigma^2_{\text{Oren-Nayar}} + 0.33} + 0.17 C_{\text{Diffuse}} \left(\frac{\sigma^2_{\text{Oren-Nayar}}}{\sigma^2_{\text{Oren-Nayar}} + 0.13} \right), \\ B &= 0.45 \frac{\sigma^2_{\text{Oren-Nayar}}}{\sigma^2_{\text{Oren-Nayar}} + 0.09}, \end{aligned}$$

and

$$\begin{aligned} s &= l \cdot v - (l \cdot n)(v \cdot n), \\ t &= \begin{cases} 1 & \text{if } s \leq 0 \\ \max(l \cdot n, v \cdot n) & \text{if } s > 0 \end{cases}, \end{aligned}$$

where C_{Diffuse} denotes the diffuse reflectance parameter in the same way as for the *Lambertian Reflector* in Equation (4), and $\sigma_{\text{Oren-Nayar}}$ is the *Oren-Nayar Roughness*, defined as the standard deviation of the diffuse microfacet angles [4, 28, 39]. While the *Oren-Nayar Model* provides significantly more control over the diffuse nature of the surface, it must be noted that it violates energy conservation for $C_{\text{Diffuse}} > 0.97$ [16, 34].

With both the specular top-layer contribution of Equation (7)-(10) and the rough diffuse secondary layer contribution of Equation (11) the complete *BRDF* for the two-layer *Surface Material* [39] can be assembled as

$$f(l, v) = f_{\text{Specular}}(l, v) + T(l, h) f_{\text{Diffuse}}(l, v), \tag{12}$$

by allowing the light transmitted through the specular top-layer to be reflected on the secondary rough diffuse layer. Thus, using the *BRDF* in Equation (12) the two-layer *Surface Material* can be specified using only a complex refractive index, $\tilde{n}_2$, and the *Beckman Roughness*, σ_{Beckman}, for the specular top-layer and a diffuse reflectance, C_{Diffuse}, and the *Oren-Nayar Roughness*, $\sigma_{\text{Oren-Nayar}}$, for the second diffuse layer, totaling in five scalar parameters.

2.3 Ambient Material

In addition to surface interactions, the *Lidar* implementation in the *AGX Sensor* module also considers the effects on the *Lidar* laser light when propagating through a medium. The medium properties for *Lidar* laser propagation through the *Sensor Environment* is specified using an *Ambient Material*. This *Ambient Material* describes the two most important effects affecting the *Lidar* laser light [2, 30, 38, 39, 47], of how the light intensity is attenuated over distance and how *Atmospheric Return*s [2, 39] appear.

The attenuation of the *Lidar* laser intensity transmitted through the medium, denoted I_T here, is described using the empirical *Beer-Lambert Law* [3, 29, 36], which is formulated as

$$I_T = Ie^{-\alpha x}, \tag{13}$$

for a light beam traveling an *Optical Path Length* of x through a medium of absorption coefficient α. This absorption coefficient α in Equation (13) used by the *Lidar* implementation contains the combined effects of both the direct energy absorption of the matter and the effects of light scattering [36, 39, 47].

In addition to direct attenuation of the *Lidar* laser light, denser atmospheric conditions such as fog, rain, snow or dust may cause a significant enough amount of the laser beam light to back-scatter in the medium to the *Lidar* detector, and may thus cause returns to be detected in the atmosphere by the *Lidar* sensor [2, 21, 30, 38, 39]. These *Atmospheric Return*s are generated at an *Optical Path Length*, x, from the medium entry point given by a *Gamma Distribution* [2, 38], defined by the probability distribution function

$$f(x, k, \theta) = \frac{1}{\Gamma(k)\theta^k} x^{k-1} e^{-x/\theta}, \tag{14}$$

given a $k > 0$ shape parameter controlling the stretching of the distribution and a $\theta > 0$ scale parameter controlling the peak magnitude of the distribution. $\Gamma(k)$ in Equation (14) is the continuous *Gamma Function*.

However, experiments observing these *Atmospheric Return*s have noted that, while the optical path length to the *Atmospheric Return*s follow the *Gamma Distribution* in Equation (14) in shape, the cumulative distribution does seldom reaches 100% [2, 30]. Thus, in addition to the absorption coefficient α and the two *Gamma Distribution* parameters k and θ, the *Ambient Material* must also be specified with an additional distribution magnitude scaling parameter, resulting in a total of four scalar parameters for the *Ambient Material*. The absorption coefficient α is generally dependent on the medium and the light wavelength, while the *Atmospheric Return* parameters are more dependent on the *Lidar* sensor itself [2, 30, 38, 39].

2.4 Beam Divergence

The laser beam emitted by a *Lidar* sensor will divergence at an angle of typically a couple of milliradians [24, 35, 39, 46, 48]. As the emitted power of the *Lidar* laser remains independent of this *Beam Divergence*, the divergence will instead cause the power of the beam to spread over a larger cross-sectional area, decreasing the intensity of the beam. However, considering that the *Lidar* detector is of a finite size, only a part of the returning beam's cross-section may land on the detector upon return, especially for targets at a distance. This effect will thus cause the power detected by the *Lidar* sensor to become attenuated with respect to the distance traveled by the laser beam [39, 48].

The actual divergence of the beam will depend on which medium the beam is propagating through. However, this is easily compensated for by using the *Optical Path Length* as distance instead of the real Euclidean length of the distance that the beam has propagated [15, 36, 39].

Usually laser beams are *Gaussian Beams*, characterized by the intensity or power distribution [15, 36, 39] of

$$I_{\text{Gaussian}}(r,x) = \frac{2P}{\pi w(x)^2} e^{-2r^2/w(x)^2}, \tag{15}$$

at radial position r from the optical axis of a beam of power P and radius w, that has propagated an *Optical Path Length* of x. The beam radius w can be described for *Gaussian Beams* using the initial waist radius and the *Rayleigh Length*, however, most *Lidar* manufacturers tend to directly specify the *Beam Divergence* angle θ_D far from the waist [24, 35, 46, 48], which allows the beam radius to be simply expressed through a cone as

$$w(x) \approx x \tan\theta_D + E, \tag{16}$$

where E denotes an offset radius that the beam has at the emitter, which is sometimes specified by manufacturers [24, 35, 46]. While the *Beam Divergence* angle θ_D and emitter radius E could be used to express the *Gaussian Beam* radius in its regular formulation, the conical approximation in Equation (16) allows specification using an unknown or zero emitter radius, and does also fit measurement data better than the full *Gaussian Beam* radius formulation [46].

The ratio of the power detected by the *Lidar* sensor, P_{Detected}, to emitted power P, as a function of *Optical Path Length* x can be found using the *Gaussian Beam* intensity in Equation (15) as

$$\frac{P_{\text{Detected}}}{P} = \frac{2}{\pi w(x)^2} \int_0^{2\pi} \int_0^{R} e^{-2\left((r-r_D)^2 \cos^2\theta + r^2 \sin^2\theta\right)/w(x)^2} r \, \mathrm{d}r \, \mathrm{d}\theta \,, \tag{17}$$

for a detector at a lateral displacement of r_D from the emitter, assumed to be on the optical axis, and of detector radius R, as illustrated in Figure 3.

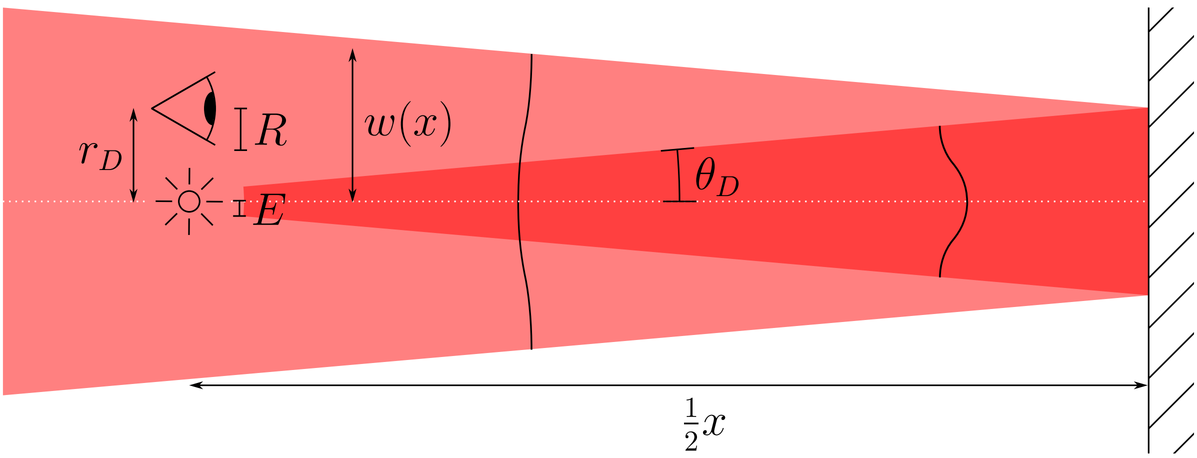

Figure 3: Illustration of the detected power of the beam attenuated due to the *Beam Divergence* $w(x)$ over an *Optical Path Length* of x with divergence angle θ_D, for a detector of radius R at radial distance r_D from the emitter of radius E. The power distribution, illustrated as Gaussian, can be seen as the black lines perpendicular to the propagation direction.

Directly analytically solving the expression in Equation (17) to arrive at an expression suitable for real-time evaluation is a fairly difficult task. Instead, starting by assuming the far field condition where

$w(x) \gg R$, then the intensity from Equation (15) in Equation (17) can be considered quasi-constant at r_D, which results in the far field approximation

$$\begin{aligned} \frac{P_{\text{DetectedFar}}}{P} &\approx \frac{1}{P} I_{\text{Gaussian}}(r_D, x) \int_0^{2\pi} \int_0^R r \, \mathrm{d}r \, \mathrm{d}\theta \\ &\approx \frac{2R^2}{w(x)^2} e^{-2r_D^2/w(x)^2}. \end{aligned} \tag{18}$$

Conversely, assuming the near field condition where $w(x) \lessapprox R$, then the expression in Equation (17) can be analytically simplified for the coaxial case where $r_D = 0$ as the approximation

$$\begin{aligned} \frac{P_{\text{DetectedNear}}}{P} &\approx \frac{2}{\pi w(x)^2} \int_0^{2\pi} \int_0^R e^{-2r^2/w(x)^2} r \, \mathrm{d}r \, \mathrm{d}\theta \\ &\approx 1 - e^{-2R^2/w(x)^2}, \end{aligned} \tag{19}$$

which can then be combined with the far field approximation in Equation (18) to form an expression combining the near field and far field behaviors.

However, just naively combining these approximations will cause the detected power ratio to become nonphysical as the *Optical Path Length* approaches zero, $x \to 0$, significantly exceeding 100%. It can be noted that it is in practice the exponential part of the far field approximation in Equation (18) that dominates in the far field, and that the fractional part is the one causing the nonphysical behavior. Using this observation, the combined but unscaled expression

$$\frac{P_{\text{Detected}}}{P} \sim \left(1 - e^{-2R^2/w(x)^2}\right) e^{-2r_D^2/w(x)^2}, \tag{20}$$

can be assembled. It is then reasonable to assume that the power ratio should be unity if the emitter and detector are coaxial and equal in size, and the *Optical Path Length* is zero,

$$\left.\frac{P_{\text{Detected}}}{P}\right|_{R=E, r_D=0, x=0} = 1,$$

which can then be applied to the combined but unscaled expression in Equation (20) to form a final expression for the attenuation due to beam divergence as

$$\frac{P_{\text{Detected}}}{P} = \frac{1}{1 - e^{-2}} \left(1 - e^{-2R^2/w(x)^2}\right) e^{-2r_D^2/w(x)^2}. \tag{21}$$

Thus, supplied with a *Beam Divergence* angle θ_D and a detector radius R, and optionally a detector offset r_D and an emitter radius E, the *Lidar* implementation in the *AGX Sensor* module uses the expression in Equation (21) to attenuate the *Lidar* beam as it propagates along its optical path.

2.5 Detection Distance Inaccuracy

In reality the medium through which the *Lidar* laser beam propagates is seldom entirely homogeneous, and even more rarely exactly the same as the medium for which the *Lidar* is calibrated. Adding to this that the internal clock of the *Lidar* sensor may not be entirely accurate, and it is reasonable that the time of flight measured and converted to a distance by the *Lidar* sensor may not always align entirely with the real distance that the *Lidar* pulse has propagated. These small errors in the measured distance are most likely to behave in a statistical manner and is therefore added by the *Lidar* implementation in the *AGX Sensor* module through a simple measurement distance perturbation with a common *Gaussian Distribution*

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2},$$

for which the *Lidar* implementation uses

$$\sigma = \sigma_{\text{Base}} + \sigma_{\text{Slope}} x,$$

to more accurately capture the growing inaccuracy at a distance resulting from cumulative inaccuracies from inhomogeneous media and the *Lidar* clock.

3 Inertial Measurement Unit

*Inertial Measurement Unit*s (*IMU*s) can come in a wide variety of different configurations, and equipped with a wide variety of different features, depending on the manufacturer and usage requirements [6, 9, 42, 45]. The *AGX Sensor* module provides an *IMU* implementation that allows modeling of most standard *IMU*s, of the kind which simply contains a set of sub-sensors and forwards their signals.

This *IMU* implementation allows sub-sensors, such as *Accelerometer*s, *Gyroscope*s or *Magnetometer*s, to be configured and placed at offsets relative to the parent *IMU* frame, in a manner similar to the 9 DoF *IMU* example in Figure 4. The *IMU* then provides a common interface through which the output format can be specified, and the output signal can be read.

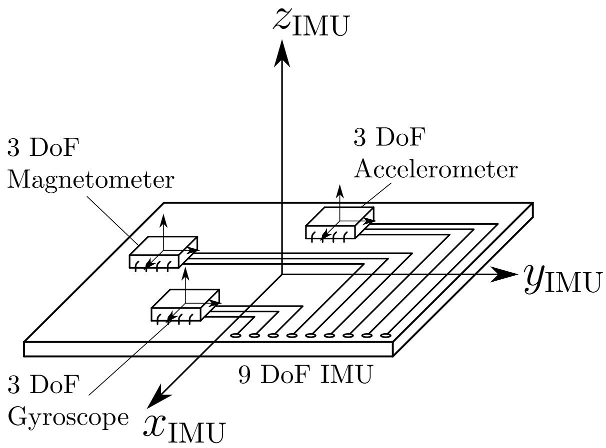

Figure 4: Example of a 9 DoF *Inertial Measurement Unit* (*IMU*) consisting of a 3 DoF *Accelerometer*, a 3 DoF *Gyroscope* and a 3 DoF *Magnetometer*, each attached at their own offsets to the parent *IMU* frame.

4 Triaxial Sensors

A *Triaxial Sensor* is in this context simply used to denote a sensor that consists of up to three output axes. For most of these sensors, each of the three output axes corresponds to each their own measurement axis, which are usually orthogonally aligned to each other to provide a three-dimensional measurement. In many cases, such sensors are simply assembled from three orthogonally aligned one dimensional sensors.

However, due to manufacturing inaccuracies the axes in these kinds of sensors may not always be properly orthogonal, and thus the sensor axes may not always align with the perceived measurement axes [40]. This can cause input along one of the measurement axes to bleed into the output of other axes, usually referred to as *Cross-axis Sensitivity* [9, 13]. The *AGX Sensor* module allows this *Cross-axis Sensitivity* to be specified through a 3-by-3 sensitivity matrix $A_{\text{Cross-axis}}$ [40] that perturbs the measured triaxial signal sample u as

$$u_{\text{Perturbed}} = A_{\text{Cross-axis}} u. \tag{22}$$

4.1 Total Noise

Many *Triaxial Sensor*s have a *Total Noise*, or *Output Noise*, specification which represents the amount of statistical noise in the output signal [44]. This noise tends to come from various sources that can be assumed to be more or less independent of the sensor state. Some of the most common types are baseline electrical and thermal noise. As these kinds of noise can be considered mostly independent, they can be assumed to follow a common *Gaussian Distribution* centered around zero

$$u_{Noise}^{(i)} \sim \mathcal{N}\left(0, \sigma_i^2\right), \tag{23}$$

given a standard deviation, or root-mean-square, of the noise σ_i for axis i as specified in the respective sensor datasheet. Using these noise values as generated from Equation (23) for each of the three axes, the *AGX Sensor* module simply adds these values to the triaxial signal to form the perturbed signal as

$$u_{\text{Perturbed}} = u + u_{Noise}.$$

4.2 Spectral Noise

In addition to *Total Noise*, many *Triaxial Sensor*s also have a *Noise Amplitude Spectral Density* specified [9, 13, 14]. For most *Triaxial Sensor*s, *Spectral Noise* tends to come mostly from electrical equipment in the sensor, especially from capacitors or amplifier circuitry, and represents the amount of additional noise that appears in the output signal at a specific bandwidth [18, 44]. *Spectral Noise* is in the general case defined as a noise where the amplitude distribution in the frequency domain follows the specified *Noise Amplitude Spectral Density*. However, directly applying the *Noise Amplitude Spectral Density* to the signal through this definition requires the entire signal to be known, something that cannot be done when each signal sample is generated in real-time and mostly independently of the previous samples.

If, instead, the underlying *Spectral Noise* source in the electrical components is assumed to be *White Noise*, mostly independent of environmental factors at a fixed frequency of interest, then the fixed-frequency noise samples can be assumed to follow a *Gaussian Distribution* like the one for *Total Noise* in Equation (23). The amplitude distribution for White Noise, however, is uniform and does thus not follow the distribution specified by the *Noise Amplitude Spectral Density*. To resolve this, the standard deviation as used in the *Gaussian Distribution* of Equation (23) can be adjusted as

$$\sigma_i = N_0^{(i)} \sqrt{\nu}, \tag{24}$$

using the *Noise Amplitude Spectral Density* root-mean-square, $N_0^{(i)}$, in axis i as specified by the respective sensor datasheet, scaled by the square-root of the bandwidth frequency ν. The noise samples for the *Spectral Noise* are thus given as

$$u_{Noise}^{(i)} \sim \mathcal{N}\left(0, \left(N_0^{(i)} \sqrt{\nu}\right)^2\right), \tag{25}$$

and when evaluated for each of the three axes is added to the triaxial signal to form the perturbed signal as

$$u_{\text{Perturbed}} = u + u_{Noise}.$$

This bandwidth frequency ν in Equation (24) and Equation (25) is in the case of the *Triaxial Sensor* signals in the *AGX Sensor* module the same as the sensor sample frequency.

4.3 Accelerometer

An *Accelerometer* is a kind of sensor which allows for the measurement of acceleration. Commercially, *Accelerometer*s tend to be available with up to three orthogonal measurement axes [5, 10, 12, 13], and are thus considered as *Triaxial Sensor*s in the *AGX Sensor* module. However, the *Accelerometer*s in the *AGX Sensor* module do support an output specification of fewer than three axes as well.

While the internals of different kinds of *Accelerometer*s can vary wildly, in the general case an *Accelerometer* can be considered as a spring-mass-damper system [17, 19, 22]. However, these model spring-mass systems of most *Accelerometer*s tend to have a very high natural frequency, of many magnitudes above the frequency at which the *AGX Dynamics* simulation runs [5, 13, 19]. As such, the *AGX Sensor* module does not model the full spring-mass-damper system, and instead relies on the dynamics simulation of the core *AGX Dynamics* to compute and provide a general *Accelerometer* reading at the frequency of the *AGX Dynamics* simulation.

The *Accelerometer* implementation in the *AGX Sensor* module attaches to a regular *AGX Dynamics* frame of reference. This frame can then be the frame of, or attach relative to the frame of, a rigid body. The *Accelerometer* then uses the nearest parent rigid body to determine the acceleration in the *Accelerometer* frame as

$$u = F_{\mathrm{A}}^{-1} \left(u_{\mathrm{RB}} + u_{\mathrm{Rotation}} + u_{\mathrm{g}} \right), \tag{26}$$

where F_{A} denotes the *Accelerometer* frame matrix, whose basis vectors can be seen illustrated in Figure 5, and $u_{\mathrm{g}} = g$ denotes the acceleration from gravity in the world frame. u_{RB} in Equation (26) is the acceleration of the nearest parent rigid body in the world frame, and is determined from the difference in the rigid body velocity

$$u_{\mathrm{RB}} = \frac{\Delta v_{\mathrm{RB}}}{\Delta t},$$

between the previous and the current *Sensor Environment* time steps.

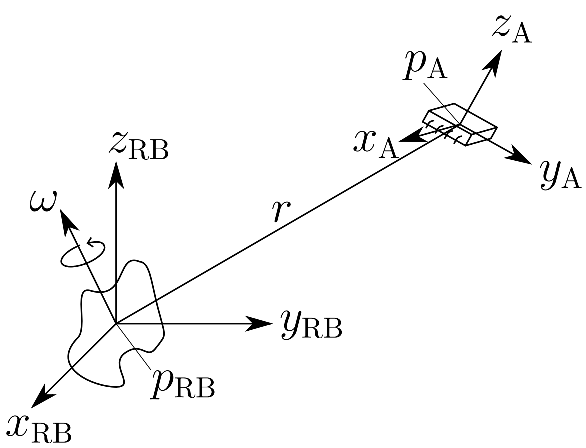

Figure 5: The *Accelerometer* as attached to a sub-frame relative to a rotating host rigid body.

The final u_{Rotation} term in Equation (26) comes from the acceleration due to rotational motion of the *Accelerometer* around the nearest parent rigid body, as illustrated in Figure 5. This acceleration is determined by evaluating the derivative of the rotation tangential velocity

$$v_{\mathrm{Rotation}} = \omega \times r, \tag{27}$$

as

$$
\begin{aligned}
u_{\text{Rotation}} &= \frac{dv_{\text{Rotation}}}{dt} \\
&= \frac{d\omega}{dt} \times r + \omega \times \frac{dr}{dt},
\end{aligned}
\tag{28}
$$

for which the angular acceleration, $d\omega/dt$, can be determined in the same way as the rigid body acceleration

$$
\frac{d\omega}{dt} \approx \frac{\Delta\omega}{\Delta t}
$$

through the difference in rotational velocity of the nearest parent rigid body between the previous and current *Sensor Environment* time steps. Meanwhile, the relative position r in Equation (28) can be determined from the positions of the nearest parent rigid body and the *Accelerometer* as

$$
r = p_{\text{A}} - p_{\text{RB}},
$$

which can be seen in Figure 5.

The remaining term, dr/dt, denoting the change in relative position in Equation (28) can be found to be the tangential velocity of Equation (27),

$$
\begin{aligned}
\frac{dr}{dt} &= v_{\text{Rotation}} \\
&= \omega \times (p_{\text{A}} - p_{\text{RB}}),
\end{aligned}
$$

as the radial distance between the nearest parent rigid body frame and the *Accelerometer* frame will remain constant over time.

The final form of Equation (26) used to compute the unperturbed *Accelerometer* signal thus becomes

$$
u = F_{\text{A}}^{-1} \left(\frac{\Delta v_{\text{RB}}}{\Delta t} + \frac{\Delta\omega}{\Delta t} \times (p_{\text{A}} - p_{\text{RB}}) + \omega \times (\omega \times (p_{\text{A}} - p_{\text{RB}})) + g \right). \tag{29}
$$

To this unperturbed signal, the *Accelerometer* implementation in the *AGX Sensor* module then applies a detectable acceleration range clamp, a *Zero g Bias* offset and the *Cross-axis Sensitivity* of Equation (22) as

$$
u_{\text{Perturbed}} = A_{\text{Cross-axis}} \left(\max\left[u_{\text{Lower}}, \min\left[u, u_{\text{Upper}} \right] \right] + u_{\text{Zero-g-Bias}} \right),
$$

to arrive at an initial perturbed signal, where u_{Lower}, u_{Upper}, $u_{\text{Zero-g-Bias}}$ and the *Cross-axis Sensitivity* specifications can be commonly found in *Accelerometer* datasheets [5, 10, 12, 13]. Other modifications, such as noise or scaling, are then added to this initial perturbed signal to add further sources of error, or reform the signal.

4.4 Gyroscope

A *Gyroscope* is a kind of sensor which allows for the measurement of rotational rate. Commercially, *Gyroscope*s tend to be available with up to three orthogonal measurement axes [9, 11, 27, 33], and are thus considered *Triaxial Sensor*s in the *AGX Sensor* module. Just like for *Accelerometer*s, the *Gyroscope*s in the *AGX Sensor* module allow for specification of output of less than three axes as well.

Internally, *Gyroscope*s can vary wildly, but in the general case a *Gyroscope* can either be considered a spinning flywheel system in one axis or a set of two *Accelerometer*s [37, 41]. These model systems of most *Gyroscope*s tend to have a very high natural frequency, of many magnitudes above the frequency at which the *AGX Dynamics* simulation runs [9, 11, 37]. As such, the *AGX Sensor* module does not model the full spinning flywheel system, and instead relies on the dynamics simulation of the core *AGX Dynamics* to compute and provide a general *Gyroscope* reading at the frequency of the *AGX Dynamics* simulation.

The *Gyroscope* implementation in the *AGX Sensor* module attaches to a regular *AGX Dynamics* frame of reference. This frame can then be the frame of, or attach relative to the frame of, a rigid body. The *Gyroscope* then uses the nearest parent rigid body to determine the rotational velocity in the *Gyroscope* frame as

$$u = F_{\mathrm{G}}^{-1}\omega, \tag{30}$$

where F_{G} denotes the *Gyroscope* frame matrix, whose basis vectors can be seen illustrated in Figure 6, and ω is the nearest parent rigid body rotation velocity in the world frame. As the resulting rotational velocity from Equation (30) is relative to the *Gyroscope* frame, each of its components also corresponds to the Euclidean rotational rate around its corresponding basis vector, and thus Equation (30) simply provides the full unperturbed *Gyroscope* signal.

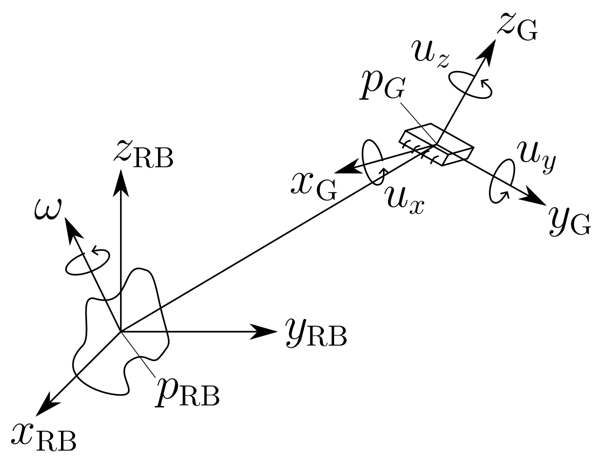

Figure 6: The *Gyroscope* as attached to a sub-frame relative to a rotating host rigid body.

To the unperturbed *Gyroscope* signal from Equation (30), the *Gyroscope* implementation in the *AGX Sensor* module then applies a detectable rotational rate range clamp, a *Zero Rate Bias* offset and the *Cross-axis Sensitivity* of Equation (22) as

$$u_{\text{Perturbed}} = A_{\text{Cross-axis}} \left(\max\left[u_{\text{Lower}}, \min\left[u, u_{\text{Upper}}\right]\right] + u_{\text{Zero-Rate-Bias}}\right),$$

to arrive at an initial perturbed signal, where u_{Lower}, u_{Upper}, $u_{\text{Zero-Rate-Bias}}$ and the *Cross-axis Sensitivity* specifications can be commonly found in *Gyroscope* datasheets [9, 11, 33]. Other modifications, such as noise or scaling, are then added to this initial perturbed signal to add further sources of error, or reform the signal.

4.4.1 Linear Acceleration Effects

One of the most significant modifications that can be added to the *Gyroscope* signal is the result of *Linear Acceleration Effects*. These effects commonly arise from mechanical inaccuracies in the manufacturing of the *Gyroscope* and are especially relevant as most *Gyroscope*s used on Earth will always be exposed to the gravitational field of Earth [49].

The implementation of *Linear Acceleration Effects* in the *AGX Sensor* module determines the acceleration on the *Gyroscope* in the same way as for an *Accelerometer* using an analogue to Equation (29) as

$$u_{\text{Acceleration}} = F_{\text{G}}^{-1}\left(\frac{\Delta v_{\text{RB}}}{\Delta t} + \frac{\Delta\omega}{\Delta t} \times (p_{\text{G}} - p_{\text{RB}}) + \omega \times (\omega \times (p_{\text{G}} - p_{\text{RB}})) + g\right),$$

using the placement configuration of the *Gyroscope* as visualized in Figure 6. This acceleration is then distributed as perturbations to the *Gyroscope* output signal axes, specified using a *Linear Acceleration Effects* matrix A_{LAE}, as

$$u_{\text{Perturbed}} = u + A_{\text{LAE}} u_{\text{Acceleration}}.$$

The *Linear Acceleration Effects* values specified in *Gyroscope* datasheets can be used to set the values in the *Linear Acceleration Effects* matrix A_{LAE} [9, 11]. In these datasheets, a single value or per-axis values will usually specify how much a linear acceleration along one of the *Gyroscope* output axis will affect its output, thus corresponding to the diagonal elements of A_{LAE}.

4.5 Magnetometer

A *Magnetometer* is a kind of sensor which allows for the measurement of *Magnetic Flux Density*. For use in machines, *Magnetometer*s tend to be available with up to three orthogonal measurement axes [26, 31], and are thus considered *Triaxial Sensor*s in the *AGX Sensor* module. Full triaxial *Magnetometer*s are usually used to determine orientation similarly to compasses. Just like for other *Triaxial Sensor*s in the *AGX Sensor* module, the *Magnetometer*s allow for specification of output of less than three axes as well.

To keep in line with other *Triaxial Sensor*s, such as *Accelerometer*s or *Gyroscope*s, the *Magnetometer* implementation samples the *Magnetic Field* at the same frequency as the main dynamics simulation of *AGX Dynamics*. The *Magnetometer* implementation attaches to a regular *AGX Dynamics* frame of reference, whose position in world space p_{M} is then used to sample the *Sensor Environment* spanning *Magnetic Field* B as

$$u = F_{M}^{-1} B\,(p_{\text{M}}),$$

to provide an unperturbed *Magnetometer* reading in the *Magnetometer* frame.

To this unperturbed signal, the *Magnetometer* implementation in the *AGX Sensor* module then applies a detectable *Magnetic Flux Density* range clamp, a *Zero Flux Bias* offset and the *Cross-axis Sensitivity* of Equation (22) as

$$u_{\text{Perturbed}} = A_{\text{Cross-axis}}\left(\max\left[u_{\text{Lower}}, \min\left[u, u_{\text{Upper}}\right]\right] + u_{\text{Zero-Flux-Bias}}\right),$$

to arrive at an initial perturbed signal, where u_{Lower}, u_{Upper}, $u_{\text{Zero-Flux-Bias}}$ and the *Cross-axis Sensitivity* specifications can be commonly found in *Magnetometer* datasheets [26, 31]. Other modifications, such as noise or scaling, are then added to this initial perturbed signal to add further sources of error, or reform the signal.

4.5.1 Magnetic Fields

As an attribute to the *Sensor Environment* an environment-spanning *Magnetic Field* can be specified, similarly as to how the *Gravity Field* is specified for the general *AGX Dynamics* simulation. This *Magnetic Field* specified for the *Sensor Environment* represents the *Magnetic Flux Density*, the B-field in units of Tesla, throughout the environment.

5 Encoder

An *Encoder* is a kind of sensor which encodes a displacement in position into a quadrature waveform. This displacement in position can either be a position displacement along a linear extension as in the case of linear *Encoder*s or a displacement in angular position as for the case of rotary *Encoder*s. The produced quadrature waveform is then commonly interpreted by a microcontroller to provide the user of the sensor with a complete value for the linear or angular position [7, 8]. The microcontroller may also assemble a value for the linear or angular speed based on the quadrature waveform.

In the *AGX Sensor* module the *Encoder* sensor provides its readings as those available after the processing by a microcontroller. To produce its signal, the *Encoder* in the *AGX Sensor* module samples the position of an unconstrained degree of freedom of a constraint between two rigid bodies in the main dynamics simulation of *AGX Dynamics*. This sampling occurs at the rate of the main dynamics simulation as to keep in line with most other sensors.

To this sampled position u the *Encoder* implementation can then apply *Gaussian Noise* as

$$\begin{aligned} u_{\text{Perturbed}} &= u + u_{Noise}, \\ u_{Noise} &\sim \mathcal{N}\left(0, \sigma^2\right), \end{aligned} \tag{31}$$

given a standard deviation, or root-mean-square, value for the noise σ to mimic minor measurement errors in the internal *Encoder* structure.

Additionally, to mimic the step-like position measurement commonly resulting from the interpretation of the discrete quadrature waveforms, the *Encoder* implementation in the *AGX Sensor* module can then apply resolution binning to the measurement position u or the perturbed signal $u_{\text{Perturbed}}$ from Equation (31) as

$$u_{\text{Binned}} = u_{\text{Resolution}} \text{floor}\left[\frac{u}{u_{\text{Resolution}}}\right]. \tag{32}$$

The *Encoder* will then operate in either an incremental *Encoder* mode, where the any of the previous sample steps of u, $u_{\text{Perturbed}}$ or u_{Binned} is wrapped within the *Encoder* measurement range as

$$u_{\text{Measurement}} = u - \left(u_{\text{Upper}} - u_{\text{Lower}}\right) \text{floor}\left[\frac{u - u_{\text{Lower}}}{u_{\text{Upper}} - u_{\text{Lower}}}\right], \tag{33}$$

or in an absolute *Encoder* mode, where any of the previous sample steps is clamped within the *Encoder* range as

$$u_{\text{Measurement}} = \max\left[u_{\text{Lower}}, \min\left[u, u_{\text{Upper}}\right]\right]. \tag{34}$$

The incremental mode of Equation (33) is used to achieve outputs resembling those from incremental *Encoder*s, such as rotary *Encoder*s without an index waveform, whereas the absolute mode of Equation (34) is used to mimic both linear and rotary *Encoder*s with absolute position measurements.

The u_{Lower}, u_{Upper} and $u_{\text{Resolution}}$ parameters in Equation (32)-(34) can be inferred from the *Encoder* geometry, mode of operation and quadrature cycle length specification commonly found in *Encoder* datasheets [7, 8].

The *Encoder* implementation in the *AGX Sensor* module also provides a measurement value for the speed, or the rate of change, of the *Encoder* position measurement. This speed value is determined similarly to how the microcontroller on a real *Encoder* would determine the speed, by measuring the time and amount of change between changes in *Encoder* measurement signal as produced by Equation (33) or Equation (34) and computing the average rate of change.

6 Odometer

An *Odometer* as a general sensor is a kind of sensor that provides the distance a machine or vehicle has traversed. *Odometer*s can in general determine this traveled distance from different sources depending on the specific type of *Odometer* that is being used. However, in this case of the *AGX Sensor* module, a plain *Odometer* refers to a direct wheel attached *Odometer* that samples wheel position from a *Wheel Encoder* and integrates the position over time. This *Odometer* is the kind that one would usually find being used in cars, machines and robots.

To determine the traveled distance measurement, the *Odometer* in the *AGX Sensor* module simply samples the position of the unconstrained rotational degree of freedom of a constraint between two rigid bodies from the main dynamics simulation of *AGX Dynamics*, at the same frequency as the main dynamics simulation. This sampled position u is then scaled by a supplied wheel radius R_{Wheel} as

$$u_{\text{Distance}} = R_{\text{Wheel}} u,$$

to provide the constraint rotational position u as a distance traveled by an imagined wheel of radius R_{Wheel}. Thus, the only parameter to specify for a plain *Odometer* becomes the wheel radius R_{Wheel} that can either be manually specified or determined from wheel geometry.

However, to allow for completeness in regard to a potential underlying *Wheel Encoder*, the *Odometer* implementation also supports resolution binning of the signal as

$$u_{\text{Binned}} = u_{\text{Resolution}} \text{floor}\left[\frac{u}{u_{\text{Resolution}}}\right].$$

given a specification of an underlying *Encoder* cycle length $u_{\text{Resolution}}$, as well as an application of *Gaussian Noise* as

$$u_{\text{Perturbed}} = u + u_{Noise},$$
$$u_{Noise} \sim \mathcal{N}\left(0, \sigma^2\right),$$

given a standard deviation, or root-mean-square, for the noise σ.

References

[1] Tomas Akenine-Möller, Eric Haines, and Naty Hoffman. *Real-Time Rendering*. 4th Edition. CRC Press, 2018.

[2] Maria Ballesta-Garcia et al. “Analysis of the performance of a polarized LiDAR imager in fog”. In: *Optics Express* 30 (Oct. 2022), p. 41524.

[3] David Blagojevic. “Weather effects on short-range LiDAR and their classification”. 2022. (Master’s Thesis).

[4] Jakub Boksansky. *Crash Course in BRDF Implementation*. Feb. 2021. URL: `https://boksajak.github.io/blog/BRDF`. (Accessed 2024-03-06).

[5] Bosch. *BMA456 - Data Sheet Digital, triaxial acceleration sensor*. (Accessed 2025-01-09).

[6] Bosch. *Small, versatile 6DoF sensor module*. (Accessed 2025-01-27).

[7] Broadcom. *AEAT-6600-T16 10-Bit to16-Bit Programmable Angular Magnetic Encoder IC*. (Accessed 2025-08-19).

[8] Broadcom. *HEDS-9040/9140 Three Channel Optical Incremental Encoder Modules*. (Accessed 2025-08-19).

[9] Analog Devices. *±1000°/Sec Precision Angular Rate Sensor*. (Accessed 2025-01-27).

[10] Analog Devices. *3-Axis, ±2 g/±4 g/±8 g/±16 g Digital Accelerometer ADXL345*. (Accessed 2025-01-09).

[11] Analog Devices. *High Stability, Low Noise Vibration Rejecting Yaw Rate Gyroscope*. (Accessed 2025-01-27).

[12] Analog Devices. *Low Noise, Wide Bandwidth, MEMS Accelerometer ADXL1003*. (Accessed 2025-01-09).

[13] Analog Devices. *Small, Low Power, 3-Axis ±16 g Accelerometer ADXL326*. (Accessed 2025-01-09).

[14] Analog Devices. *Tactical Grade, Ten Degrees of Freedom Inertial Sensor*. (Accessed 2025-01-27).

[15] S.J. Frank L. Pedrotti and Leno S. Pedrotti. *Introduction to Optics*. 2nd Edition. Prentice-Hall, 1993.

[16] Yasuhiro Fujii. *A tiny improvement of Oren-Nayar reflectance model*. 2024. URL: `https://mimosa-pudica.net/improved-oren-nayar.html`. (Accessed 2024-03-08).

[17] Zine Ghemari and Salah Saad. “Enhancement of capacitive accelerometer operation by parameters improvement”. In: *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields* 32.3 (2019), e2568.

[18] Zine Ghemari and Salah Saad. “Parameters improvement and suggestion of new design of capacitive accelerometer”. In: *Analog Integrated Circuits and Signal Processing* 92 (2017), pp. 443–451.

[19] Zine Ghemari, Saad Salah, and Rabah Bourenane. “Resonance effect decrease and accuracy increase of piezoelectric accelerometer measurement by appropriate choice of frequency range”. In: *Shock and Vibration* 2018.1 (2018), p. 5370438.

[20] David J. Griffiths. *Introduction to Electrodynamics*. 4th Edition. Cambridge University Press, 2019.

[21] Martin Hahner et al. “Fog Simulation on Real LiDAR Point Clouds for 3D Object Detection in Adverse Weather”. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 15263–15272.

[22] Faezeh Arab Hassani, Amir Farrokh Payam, and Morteza Fathipour. “Design of a smart MEMS accelerometer using nonlinear control principles”. In: *Smart structures and systems* 6.1 (2010), pp. 1–16.

[23] Eric Heitz. "Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs". In: *Journal of Computer Graphics Techniques (JCGT)* 3.2 (June 2014), pp. 48–107.

[24] Hesai. *PandarQT 64-Channel Short-Range Mechanical LiDAR User Manual*. (Accessed 2024-02-05).

[25] Naty Hoffman. "Physics and Math of Shading". In: *Siggraph*. 2015.

[26] Texas Instruments. *TMAG5170 High-Precision 3D Linear Hall-Effect Sensor With SPI*. (Accessed 2025-01-31).

[27] TDK InvenSense. *High Performance Automotive Gyroscope*. (Accessed 2025-01-27).

[28] Sébastien Lagrade and Charles de Rousiers. "Moving Frostbite to Physically based rendering". In: Siggraph 2014. DICE, 2014.

[29] Johann Heinrich Lambert. *Photometria sive de mensura et gradibus luminis, colorum et umbrae*. Augustae Vindelicorum, 1760.

[30] Clemens Linnhoff et al. "Measuring the Influence of Environmental Conditions on Automotive Lidar Sensors". In: *Sensors (Basel)* 1 (2022), p. 5266.

[31] Melexis. *MLX90395 Triaxis Magnetometer Node*. (Accessed 2025-01-31).

[32] Mathias Müller. *LiDAR Explained – Understanding LiDAR Specifications and Performance*. 2020. URL: `https://www.blickfeld.com/blog/understanding-lidar-specifications/`. (Accessed 2024-01-23).

[33] muRata. *6-DOF Gyroscope and Accelerometer with Digital SPI Interface*. (Accessed 2025-01-27).

[34] Michael Oren and Shree K. Nayar. "Generalization of Lambert's reflectance model". In: Siggraph 1994. New York, NY, USA: Association for Computing Machinery, 1994, pp. 239–246.

[35] Ouster. *OS1 Mid-Range High-Resolution Imaging Lidar Datasheet*. 2023. (Accessed 2024-02-06).

[36] Rüdiger Paschotta. *Encyclopedia of Laser Physics and Technology*. [Physical Copy]. Wiley-VCH, 2008. (Accessed 2024-01-30).

[37] Chandradip Patel and Patrick McCluskey. "Modeling and simulation of the MEMS vibratory gyroscope". In: *13th InterSociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*. 2012, pp. 928–933.

[38] R. Rasshofer, M. Spies, and H. Spies. "Influences of weather phenomena on automotive laser radar systems". In: *Advances in Radio Science* 9 (July 2011).

[39] Philip Rosberg. "Simulation of real-time Lidar sensor in non-ideal environments: Master's Thesis in Engineering Physics". 2024.

[40] M Šipoš, J Roháč, and P Nováček. "Improvement of electronic compass accuracy based on magnetometer and accelerometer calibration". In: *Acta Physica Polonica A* 121.4 (2012), pp. 945–949.

[41] Sara Stančin and Sašo Tomažič. "On the interpretation of 3D gyroscope measurements". In: *Journal of Sensors* 2018.1 (2018), p. 9684326.

[42] STMicroelectronics. *iNEMO inertial module: 3D accelerometer, 3D gyroscope, 3D magnetometer*. (Accessed 2025-01-27).

[43] Synopsys. *What is LiDAR and How Does it Work?* 2020. URL: `https://www.synopsys.com/glossary/what-is-lidar.html`. (Accessed 2024-01-23).

[44] Wilcoxon Sensing Technologies. *Understanding the accelerometer noise specification*. (Accessed 2025-05-28).

[45] Xsens Technologies. *MTi 600-series Datasheet*. (Accessed 2025-01-27).

[46] Velodyne. *VLP-16 User Manual 63-9243 Rev. E*. 2019. (Accessed 2024-01-24).

[47] Johan Viklund. "Atmospheric Attenuation for Lidar Systems in Adverse Weather Conditions". 2021. (Master's Thesis).

[48] Kevin Vincent. *How does Lidar work? (in detail)*. 2023. URL: `https://www.outsight.ai/insights/how-does-lidar-work`. (Accessed 2024-01-23).

[49] Harvey Weinberg. "Gyro mechanical performance: The most important parameter". In: *Technical Article MS-2158* (2011), pp. 1–5.