

Parametric Design and Evolutionary Optimisation of Gray Code Optical Encoders for Marine Rudder Position Sensing

Susanne Coates
Independent Researcher
`contact@susannecoates.com`

November 16, 2025

Abstract

We present a parametric design system for generating 3D-printable Gray code optical encoders optimised for absolute position sensing in marine autopilot applications. The system addresses the challenge of translating Gray code's mathematical elegance into manufacturable physical artefacts whilst satisfying competing constraints of additive manufacturing limitations, optical sensor requirements, and mechanical robustness. Through evolutionary optimisation using genetic algorithms, the system explores multi-dimensional parameter spaces to identify near-optimal configurations balancing printability, resolution, encoding efficiency, and manufacturability. The implementation separates concerns through modular architecture: mathematical abstractions in the Gray code subsystem, geometric realisation through constructive solid geometry, constraint validation across multiple domains, and automated optimisation via tournament selection with elitism. Empirical results demonstrate that the genetic algorithm successfully navigates complex constraint interactions, yielding encoder designs with fitness scores of 1.115–1.140 across 50 generations. The system generates OpenSCAD geometries manufacturable on standard fused deposition modelling printers with 0.4 mm nozzles, providing absolute position encoding for sailboat rudder control systems with angular resolutions of 0.117–0.5625 degrees per position. The parametric approach enables rapid adaptation to varying dimensional constraints, resolution requirements, and manufacturing capabilities without fundamental redesign.

1 Introduction

1.1 Motivation and Context

Autonomous marine navigation systems require precise knowledge of rudder position for effective course control. In sailboat autopilot applications, position sensing must operate reliably in environments characterised by vibration, moisture, electrical noise, and limited access to replacement components. Traditional sensing technologies each present distinct limitations that compromise their suitability for this application domain. Mechanical rheostats, whilst simple in principle, rely on sliding contact between a resistive

element and a wiper arm. This contact inevitably wears over time, and in marine environments where moisture accelerates corrosion, the reliability concerns become particularly acute. Inductive encoders offer non-contact operation but require complex drive electronics to generate and measure changing inductance values, increasing system complexity and potential failure modes. Incremental optical encoders provide high resolution but lose position information during power cycles, necessitating homing procedures that are fundamentally incompatible with safety-critical marine control systems where the vessel’s heading must be maintained through power fluctuations.

Absolute optical encoders address these limitations by providing immediate position knowledge upon power application without any initialisation sequence. Each discrete angular position corresponds to a unique binary code pattern, eliminating the cumulative error inherent in incremental sensing and the reliability concerns of contact-based methods. However, the straightforward application of direct binary encoding introduces a critical vulnerability in physical implementations. During certain transitions, multiple bits must change simultaneously—most dramatically illustrated by the transition from position 127 to 128 in 8-bit binary encoding, which requires all eight bits to flip from 1 to 0 or vice versa. In real-world mechanical systems where perfect alignment is unattainable and sensor response times vary, these multi-bit transitions create temporal windows during which intermediate, invalid code patterns may be read. A control system sampling position during such a transition might observe arbitrary erroneous values, potentially causing dangerous responses in safety-critical applications.

Gray code [1] elegantly resolves this fundamental problem through its defining mathematical property: any two adjacent positions in the sequence differ by exactly one bit. This single-bit-change characteristic transforms the multi-bit transition vulnerability into a bounded error condition. Even with significant mechanical misalignment or sensor timing skew, the worst-case error becomes a one-position offset rather than an arbitrary jump to an invalid code. For marine applications where mechanical precision is inherently limited by the vessel’s motion and environmental conditions remain harsh, Gray code’s error-resistance properties prove essential rather than merely desirable.

1.2 Design Challenge

Whilst Gray code’s mathematical properties have been well-established since the 1950s, translating these abstractions into manufacturable physical encoders presents substantial engineering challenges that extend far beyond the mathematics. The design problem is fundamentally multi-objective: we must simultaneously satisfy manufacturing constraints imposed by additive fabrication processes, optical requirements for reliable sensor discrimination, geometric constraints arising from mounting hardware, and mechanical robustness requirements for operational reliability.

Fused deposition modelling, the most accessible additive manufacturing technology for onboard fabrication, imposes hard limits on achievable feature sizes. The thermoplastic extrusion process through heated nozzles of finite diameter—typically 0.4 mm for standard applications—establishes lower bounds on feature dimensions. Reliable feature formation requires widths approaching two nozzle diameters to permit proper fusion between adjacent perimeter passes. Gaps between features must similarly exceed minimum thresholds; insufficient gap width permits unintended bridging where extruded material spans voids that should remain transmissive to optical sensing. These manufacturing constraints are not soft preferences but hard boundaries: violations result in unusable

artefacts rather than merely suboptimal performance.

Optical sensing requirements interact complexly with geometric design. Sensors require sufficient arc length to reliably discriminate between transmissive and opaque regions, particularly as track radius decreases toward the mounting centre. The Gray code mathematical structure demands concentric circular tracks, with each bit position mapped to a specific radius. The least significant bit, exhibiting the highest transition frequency, naturally occupies the outermost track where arc lengths are maximised. However, the available radial space between outer diameter and inner mounting hole must accommodate all tracks with appropriate widths and inter-track spacing. Track count scales logarithmically with desired position resolution, creating a fundamental tension: higher resolution demands more tracks competing for limited radial space, forcing dimensional compromises that may violate manufacturing or optical constraints.

The interactions between these constraints are decidedly non-linear. Aggressive dimensional reduction—narrower tracks, tighter spacing, smaller gaps—permits higher track counts and improved resolution within constrained radial space, but approaches or exceeds manufacturing capability limits. Conservative sizing provides comfortable printability margins but limits achievable resolution or forces overall dimensions beyond mounting constraints. Manual parameter tuning through iterative physical prototyping proves time-consuming and unlikely to identify optimal configurations in multi-dimensional parameter spaces where objectives compete and constraints interact.

1.3 Contributions

This work addresses the encoder design challenge through evolutionary optimisation operating on a parametric design space, yielding several contributions to the practical realisation of Gray code optical encoders. We present a modular software architecture that cleanly separates mathematical abstractions of Gray code theory from geometric realisation and manufacturing constraint validation. This separation enables independent development and testing of components whilst maintaining clear interfaces between subsystems. The architecture supports both interactive design through graphical interfaces and automated batch processing for parameter space exploration.

Central to our approach is a multi-objective fitness function that encodes engineering knowledge about manufacturing feasibility, optical performance, and mechanical robustness. Rather than requiring manual navigation of competing objectives, the fitness function quantifies design quality across multiple dimensions, enabling automated search algorithms to identify high-quality parameter combinations. We demonstrate empirically that genetic algorithms successfully navigate these complex constraint landscapes, consistently improving fitness scores from initial random populations to converged solutions that substantially exceed manually-tuned designs.

The implementation generates OpenSCAD geometry files manufacturable on standard desktop FDM printers without requiring industrial fabrication capabilities. This accessibility extends the utility of custom encoder design beyond specialised facilities to include onboard fabrication during extended voyages where access to replacement components may be severely limited. Validation confirms that evolved parameter sets satisfy all manufacturing constraints whilst providing angular resolutions suitable for marine rudder control applications.

The system has been developed within the context of SCANS (Sérénité’s Control Automation and Navigation System), a distributed ROS-2-based autopilot architecture for

a 1984 Jeanneau Sun Fizz sailboat. However, the parametric approach generalises to arbitrary dimensional constraints, resolution requirements, and manufacturing capabilities. The open-source implementation facilitates adaptation to diverse position sensing applications beyond the originating marine context.

2 Materials and Methods

2.1 Gray Code Theory and Implementation

2.1.1 Mathematical Foundation

For any position n in the range $[0, N - 1]$ where N represents the total position count, Gray code conversion from binary employs the exclusive-OR operation:

$$\text{Gray}(n) = n \oplus (n \gg 1) \quad (1)$$

This bitwise XOR between the position value and its right-shifted counterpart ensures the single-bit-change property. Individual bit extraction for an m -bit code follows:

$$\text{bit}_i = \left(\left\lfloor \frac{\text{Gray}(n)}{2^i} \right\rfloor \right) \bmod 2 \quad (2)$$

where $i \in [0, m - 1]$ and $m = \lceil \log_2(N) \rceil$.

The implementation in `gray_code/converter.py` provides core operations:

```

1 def binary_to_gray(n: int) -> int:
2     """Convert binary position to Gray code"""
3     return n ^ (n >> 1)
4
5 def gray_code_bits(pos: int, num_bits: int) -> List[int]:
6     """Extract bit values for Gray-encoded position"""
7     gray_value = binary_to_gray(pos)
8     return [(gray_value >> i) & 1 for i in range(num_bits)]

```

Listing 1: Gray code conversion implementation

2.1.2 Track Pattern Generation

Individual bits map to concentric physical tracks on the encoder disc. Track 0 (outermost radius) carries the least significant bit (LSB), whilst Track $m - 1$ (innermost radius) carries the most significant bit (MSB). This radial ordering maximises arc length for the highest-frequency transitions (LSB alternates nearly every position) whilst accepting shorter arc lengths for the lowest-frequency transitions (MSB changes once per half-cycle).

For each track, the bit pattern across all N positions forms a binary sequence $[b_0, b_1, \dots, b_{N-1}]$ where $b_i \in \{0, 1\}$. Rather than generating individual apertures for each 1-bit, the algorithm identifies maximal contiguous subsequences of 1s and creates single continuous cutouts spanning the corresponding angular ranges. This reduces geometric complexity and improves manufacturing reliability.

2.2 Geometric Generation

2.2.1 Track Layout

Each track occupies an annular sector defined by inner radius $R_{\text{inner},i}$, outer radius $R_{\text{outer},i}$, and angular extent θ_{arc} . Given m tracks with uniform track width w and inter-track spacing s , the track pitch $p = w+s$ establishes centre-to-centre distance. Track i (indexed from 0 at outermost) occupies radial bounds:

$$R_{\text{outer},i} = R_{\text{outer}} - i \cdot p \quad (3)$$

$$R_{\text{inner},i} = R_{\text{outer},i} - w \quad (4)$$

The fundamental geometric constraint requires that the innermost track not extend below the mounting hole:

$$R_{\text{inner},m-1} \geq R_{\text{inner,mount}} \quad (5)$$

2.2.2 Arc Sector Generation

For positions distributed over arc angle θ_{arc} , each position corresponds to angular increment $\Delta\theta = \theta_{\text{arc}}/N$. A contiguous run of k consecutive 1-bits beginning at position p generates a cutout spanning angular interval $[p \cdot \Delta\theta, (p+k) \cdot \Delta\theta]$.

Arc boundary points are computed parametrically:

$$x(\theta) = R \cos(\theta) \quad (6)$$

$$y(\theta) = R \sin(\theta) \quad (7)$$

with θ interpolated over 50 discrete steps for smooth approximation. The `geometry/arc_utils.py` module implements these primitives using SolidPython [2], generating OpenSCAD code through constructive solid geometry operations.

2.3 Manufacturing Constraints

2.3.1 FDM Limitations

Fused deposition modelling extrudes thermoplastic through heated nozzles operating at diameters determined by application requirements: 0.4 mm for standard work, 0.16 mm for precision applications. The nozzle diameter establishes a fundamental lower bound on achievable feature sizes. Reliable feature formation requires widths approaching two nozzle diameters, approximately 0.8 mm for 0.4 mm nozzles, to permit two perimeter passes that properly fuse into coherent structures. Features narrower than this threshold may print as disconnected filament traces or fail to form entirely, rendering the encoder unusable.

Inter-feature gaps present analogous constraints. Gaps narrower than approximately 0.5 mm risk unintended bridging where extruded material spans voids that should remain transmissive to optical sensing. The printer's motion control system cannot position the nozzle with arbitrarily fine precision; mechanical backlash, thermal expansion, and vibration introduce positioning uncertainties on the order of 0.1 mm. Gaps approaching or

exceeding these uncertainty bounds may close partially or fully during printing, occluding regions that must remain clear for correct optical encoding.

Gap width expressed in angular measure θ_{gap} translates to linear dimension varying with radius according to:

$$d_{\text{gap}} = \frac{\theta_{\text{gap}} \cdot \pi \cdot R}{180} \quad (8)$$

At the outer radius where arc lengths are maximised, a given angular gap corresponds to the longest linear dimension. Moving inward toward the mounting centre, the same angular measure corresponds to progressively shorter linear dimensions. This radial dependence creates a fundamental tension: outer tracks naturally satisfy minimum gap width constraints whilst inner tracks approach or violate these constraints, potentially requiring gap width increases that reduce the number of distinct angular positions representable within the arc angle constraint.

2.3.2 Constraint Validation

The system implements validation across three distinct domains, each addressing different aspects of design correctness. Parameter validation, implemented in `utils/parameters.py`, checks basic geometric consistency including positive dimensions, valid angular ranges, and verification that all tracks fit within available radial space. This validation catches fundamental configuration errors before expensive geometric generation proceeds. Gray code validation, residing in `gray_code/validator.py`, verifies that the generated encoding patterns exhibit the single-bit-change property between all adjacent positions. It analyses individual track characteristics including transition counts, run lengths representing maximal contiguous sequences of identical bits, and pattern balance indicating approximately equal counts of 0s and 1s. Finally, printability analysis through `utils/printer_constraints.py` evaluates minimum feature dimensions, gap widths, and track spacing against the FDM capabilities specified in the configuration. Only designs passing all three validation stages proceed to geometric generation and OpenSCAD file export.

2.4 Evolutionary Optimisation

2.4.1 Genetic Algorithm Architecture

The genetic algorithm explores the parameter space to identify configurations satisfying multiple competing objectives. Each candidate design (genome) represents a complete parameter vector:

$$\mathbf{g} = (d_{\text{outer}}, d_{\text{inner}}, t, \theta_{\text{arc}}, N, m, w, s, \theta_{\text{gap}}, \dots) \quad (9)$$

For constrained optimisation (e.g., fixed mounting geometry), certain parameters remain constant whilst others vary. The rudder encoder application fixes d_{outer} , d_{inner} , and θ_{arc} to mounting constraints, optimising only track layout parameters (w , s , θ_{gap}).

2.4.2 Fitness Function

The multi-objective fitness function computes a weighted sum of normalised component scores according to:

$$f(\mathbf{g}) = \sum_i w_i \cdot f_i(\mathbf{g}) \quad (10)$$

The printability component f_{print} (weighted at 40%) evaluates manufacturability through systematic minimum feature analysis across all tracks. Invalid designs that violate hard manufacturing constraints receive $f_{\text{print}} = 0$, immediately eliminating them from reproduction. Valid designs score based on their margin above minimum thresholds: designs providing comfortable safety margins score near 1.0, whilst those barely satisfying constraints score proportionally lower. This scoring approach rewards robustness to manufacturing variations rather than merely accepting marginal validity.

The resolution component f_{res} (20%) quantifies angular discrimination capability. Higher position counts directly improve resolution, but the relationship exhibits diminishing returns beyond certain thresholds. A 64-position encoder provides only modest resolution improvement over 32 positions (factor of 2) whilst requiring an additional track, consuming radial space and increasing manufacturing complexity. The scoring function captures this non-linearity by penalising both insufficient resolution for the application and excessive resolution beyond practical benefit.

Encoding efficiency f_{eff} (20%) measures Gray code utilisation as the ratio $\eta = N/2^m$ where N represents actual position count and m represents track count. An encoder using 32 positions with 5 tracks achieves $\eta = 32/32 = 1.0$, utilising all available codes. Configurations using fewer positions than 2^m exhibit fractional efficiency, indicating unused code space. Whilst this unused capacity imposes no direct performance penalty, it suggests suboptimal design choices that waste resources.

Size optimisation f_{size} (10%) rewards proximity to target dimensions specified by mounting constraints. Substantial deviation in either direction indicates problems: undersized designs fail to utilise available space and may compromise track dimensions unnecessarily, whilst oversized designs violate mounting constraints and cannot be installed. The scoring function penalises deviations proportionally, creating selection pressure toward dimensional accuracy.

Manufacturability f_{mfg} (10%) captures secondary considerations affecting practical usability beyond the explicit printability constraints. This component prefers standard dimensional values that align with common measurement increments, reasonable arc angles that simplify sensor positioning, and overall design simplicity affecting assembly and optical alignment procedures. These factors prove difficult to quantify precisely but nevertheless influence real-world success rates.

Genomes satisfying all validation constraints receive a 20% multiplicative bonus through:

$$f_{\text{final}}(\mathbf{g}) = \begin{cases} 1.2 \cdot f(\mathbf{g}) & \text{if valid} \\ f(\mathbf{g}) & \text{otherwise} \end{cases} \quad (11)$$

This bonus creates a fitness cliff separating valid from invalid designs, ensuring that evolution strongly favours feasible solutions even when invalid configurations might otherwise score well on individual components.

2.4.3 Evolutionary Operators

Initialisation: Population of 30–50 genomes generated through uniform random sampling within parameter bounds. For constrained optimisation, fixed parameters adopt specified values whilst optimisable parameters vary randomly.

Selection: Tournament selection with tournament size $k = 3$. Small subsets of the population compete; highest-fitness individual from each tournament becomes a parent. This approach maintains selection pressure whilst preserving diversity better than roulette wheel or rank-based methods.

Elitism: Top 5% of each generation transfers unchanged to the next generation, preventing loss of high-quality solutions during stochastic operations.

Crossover: Uniform crossover for optimisable parameters. Each parameter inherits from one parent or the other with equal probability. Fixed parameters copy from either parent (both hold identical values).

Mutation: Each optimisable parameter mutates independently with probability $p_m = 0.3\text{--}0.4$. Mutations apply bounded random perturbations:

$$x' = \text{clip}(x \cdot \alpha, x_{\min}, x_{\max}) \quad (12)$$

where $\alpha \sim \text{Uniform}(0.5, 1.5)$ provides $\pm 50\%$ variation and clip enforces parameter-specific bounds.

Fitness Sharing: To maintain population diversity and prevent premature convergence, fitness sharing penalises genomes in crowded regions of parameter space. For each genome i , a niche count accumulates based on parameter distances to other genomes:

$$n_i = \sum_{j \neq i} \begin{cases} 1 - \frac{d_{ij}}{\sigma} & \text{if } d_{ij} < \sigma \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

where d_{ij} represents normalised Euclidean distance between genomes i and j in parameter space, and $\sigma = 0.5$ defines the sharing radius. Adjusted fitness becomes:

$$f'_i = \frac{f_i}{1 + 0.2 \cdot n_i} \quad (14)$$

This penalisation encourages exploration of under-represented parameter regions.

2.4.4 Algorithm Implementation

The genetic algorithm executes for a fixed generation count (typically 50) without early termination criteria. This ensures thorough exploration rather than premature convergence to local optima. Algorithm pseudocode:

2.5 Software Architecture

The implementation adopts a layered architecture that maintains clear separation of concerns across subsystems. The Gray code subsystem, residing in `gray_code/`, implements mathematical operations entirely independently of geometric considerations. This module provides conversion functions translating binary positions to Gray-encoded values, bit extraction operations identifying individual track patterns, pattern validation verifying the single-bit-change property, and sequence analysis computing transition statistics. The subsystem's independence from geometry enables unit testing against known Gray code properties without invoking the geometric generation pipeline.

Geometric generation, implemented in `geometry/`, builds upon these mathematical foundations to produce three-dimensional solid models. This subsystem uses the SolidPython library to generate OpenSCAD code through constructive solid geometry

Algorithm 1 Genetic Algorithm for Encoder Optimisation

```
1: Initialise population  $P$  of size  $n$  with random genomes
2: for  $g = 1$  to  $G$  do
3:   for each genome  $\mathbf{g} \in P$  do
4:     Evaluate  $f(\mathbf{g})$  using multi-objective fitness function
5:   end for
6:   Apply fitness sharing to adjust  $f(\mathbf{g}) \rightarrow f'(\mathbf{g})$ 
7:   Sort  $P$  by  $f'(\mathbf{g})$  descending
8:    $P_{\text{next}} \leftarrow$  top 5% of  $P$  {Elitism}
9:   while  $|P_{\text{next}}| < n$  do
10:    Select parents  $\mathbf{g}_1, \mathbf{g}_2$  via tournament selection
11:    Generate offspring  $\mathbf{g}'_1, \mathbf{g}'_2$  via crossover
12:    Apply mutation to  $\mathbf{g}'_1, \mathbf{g}'_2$ 
13:     $P_{\text{next}} \leftarrow P_{\text{next}} \cup \{\mathbf{g}'_1, \mathbf{g}'_2\}$ 
14:  end while
15:   $P \leftarrow P_{\text{next}}$ 
16: end for
17: return Best genome from  $P$ 
```

operations. Arc primitives create circular sector boundaries through parametric point generation, track pattern generation transforms binary sequences into physical cutouts, and Boolean operations combine solid components with transmissive voids to produce complete encoder disc models. The geometric subsystem requires only bit patterns and dimensional parameters as inputs, remaining decoupled from the mathematical details of Gray code generation.

Parameter management through `utils/` encapsulates all configuration information in the `EncoderParameters` dataclass. This subsystem implements comprehensive validation checking geometric consistency, Gray code correctness, and manufacturing feasibility. The printability analyser evaluates dimensional constraints without coupling to either Gray code mathematics or geometric generation logic. This separation enables validation rule modifications—for example, adjusting minimum feature sizes for different nozzle diameters—without affecting other system components.

The optimisation subsystem in `genetic_optimizer.py` orchestrates fitness evaluation by invoking the validation subsystems, maintains population state across generations, and executes evolutionary operators including selection, crossover, and mutation. Critically, the optimiser operates entirely through the parameter subsystem interface, never directly manipulating Gray code or geometric objects. This abstraction permits alternative optimisation algorithms—simulated annealing, particle swarm optimisation, or gradient-based methods if fitness landscapes prove sufficiently smooth—to substitute without modifying validation or generation logic.

Finally, the generation pipeline coordinated by `encoder_generator.py` provides command-line access to the complete system. This interface handles parameter loading from predefined configurations or custom JSON files, orchestrates validation across all subsystems, invokes geometric generation upon successful validation, and exports OpenSCAD files suitable for visualisation and STL conversion. The pipeline architecture supports both interactive design iteration and automated batch processing for parameter space exploration or sensitivity analysis.

3 Results

3.1 Genetic Algorithm Performance

3.1.1 Convergence Behaviour

Figure ?? shows fitness evolution over 50 generations for the rudder encoder optimisation (population size $n = 30$, constrained to fixed outer diameter 100 mm, inner diameter 35 mm, arc angle 90°, 32 positions). Initial population exhibits fitness range 0.406–0.581 with mean 0.413. The algorithm identifies substantially improved configurations within the first generation, achieving fitness 0.859. Continued evolution yields further improvements: generation 5 (0.974), generation 16 (1.105), generation 47 (1.135), converging to 1.139 by generation 50.

The fitness trajectory demonstrates active exploration rather than immediate convergence. Unlike pre-fitness-sharing implementations where optimal solutions appeared in generation 1 and persisted unchanged through all subsequent generations, the modified algorithm exhibits continued fitness improvements across multiple generations. This behaviour indicates successful navigation of the constraint landscape rather than premature convergence to local optima.

3.1.2 Parameter Evolution

Table 1 presents parameter values for select generations. The optimised solution (generation 48) specifies track width 5.30 mm, track spacing 0.97 mm, and gap width 3.23°. These values differ substantially from intuitive “round number” choices (e.g., 3 mm track width, 1 mm spacing, 3° gaps), suggesting the genetic algorithm successfully identified non-obvious parameter combinations satisfying the multi-objective fitness function.

Table 1: Parameter evolution across selected generations

Generation	Fitness	Track Width (mm)	Spacing (mm)	Gap Width (°)
0 (initial)	0.581	2.67	0.46	3.72
1	0.859	2.67	2.35	4.95
5	0.974	2.86	1.13	6.00
16	1.105	4.08	3.00	1.86
47	1.135	1.67	3.00	2.29
48 (best)	1.139	5.30	0.97	3.23

3.1.3 Population Diversity

The fitness sharing mechanism maintains population diversity throughout evolution. Examining the top five genomes at generation 48 reveals parameter ranges: track width [0.51–5.30] mm, spacing [0.46–3.00] mm, gap width [2.25–3.23]°. This diversity contrasts sharply with pre-fitness-sharing populations where elitism and strong selection pressure caused rapid convergence to near-identical genomes differing only through minor mutations.

3.2 Validation Results

3.2.1 Manufacturing Feasibility

The optimised design satisfies all manufacturing constraints for standard FDM printing (0.4 mm nozzle):

- **Minimum feature size:** Track width 5.30 mm \gg 0.8 mm threshold. All tracks exceed minimum requirements.
- **Minimum gap width:** At outer radius 50 mm, gap width 3.23° corresponds to linear dimension 2.82 mm \gg 0.5 mm threshold. Inner tracks exhibit larger angular spans, further exceeding requirements.
- **Track spacing:** 0.97 mm spacing provides adequate clearance for perimeter separation, exceeding the 0.5 mm minimum.
- **Wall thickness:** Base disc thickness 2.3 mm supports three perimeter passes at 0.4 mm line width (1.2 mm) with remaining material for structural integrity.

The `PrintabilityAnalyzer` reports zero violations and zero warnings for this parameter set.

3.2.2 Gray Code Correctness

The `GrayCodeValidator` confirms pattern correctness across all validation criteria:

- **Single-bit transitions:** All adjacent positions differ by exactly one bit. No multi-bit transitions observed.
- **Track balance:** Each track exhibits approximately equal counts of 0s and 1s (16:16 for 32-position encoder), indicating balanced optical duty cycles.
- **Transition distribution:** LSB (Track 0) exhibits 16 transitions, MSB (Track 4) exhibits 1 transition, following expected frequency scaling.
- **Maximum run length:** Longest contiguous sequence of identical bits on any track: 16 positions. This determines the maximum arc length requiring continuous solid material or void.

3.2.3 Geometric Validation

The complete disc assembly satisfies geometric constraints:

- **Radial clearance:** Innermost track (Track 4) inner radius 39.88 mm $>$ mounting hole radius 17.5 mm. Adequate clearance prevents track interference with mounting hardware.
- **Track fitting:** Five tracks with pitch 6.27 mm ($5.30\text{ mm} + 0.97\text{ mm}$) require 31.35 mm radial space. Available space ($50\text{ mm} - 17.5\text{ mm} = 32.5\text{ mm}$) accommodates all tracks with margin.
- **Arc validity:** 90° arc angle provides adequate angular extent for optical sensor positioning. Sensors placed at track mid-radii with angular spacing $\Delta\theta/2$ avoid boundary regions where transitions occur.

3.3 Comparison with Manual Design

Table 2 compares the genetically-optimised design against a manually-tuned baseline. The manual design employed intuitive parameter choices (3 mm track width, 1 mm spacing, 3° gaps). Whilst the manual design passes validation, its fitness score (1.024) falls substantially below the optimised design (1.139), primarily due to suboptimal printability margins and reduced encoding efficiency.

Table 2: Comparison of manual versus genetically-optimised designs

Parameter	Manual Design	GA-Optimised
Track Width (mm)	3.00	5.30
Track Spacing (mm)	1.00	0.97
Gap Width (°)	3.00	3.23
Fitness Score	1.024	1.139
Printability Score	0.85	1.00
Resolution Score	0.80	0.80
Efficiency Score	1.00	1.00
Size Score	1.00	1.00
Manufacturability	0.85	0.90

The improvement derives primarily from the GA’s ability to identify parameter combinations maximising printability margins whilst remaining within geometric constraints. The wider track width (5.30 mm versus 3.00 mm) and slightly wider gap (3.23° versus 3.00°) provide superior reliability margins for FDM manufacturing whilst the reduced spacing (0.97 mm versus 1.00 mm) recovers radial space.

3.4 Adaptability to Alternative Specifications

To demonstrate parametric adaptability, we applied the optimiser to three alternative scenarios:

1. **High-resolution full-rotation encoder:** 64 positions, 120 mm outer diameter, 30 mm inner diameter, 360° arc. Optimiser yielded fitness 1.087 with 6 tracks, 4.1 mm track width, 1.3 mm spacing, 2.1° gaps. Angular resolution: 5.625° per position.
2. **Compact partial-rotation encoder:** 16 positions, 80 mm outer diameter, 20 mm inner diameter, 90° arc. Optimiser achieved fitness 1.152 with 4 tracks, 6.8 mm track width, 1.1 mm spacing, 3.7° gaps. Angular resolution: 5.625° per position (22.5° rudder travel range mapped to 90° encoder rotation via 4:1 gearing).
3. **Precision encoder with 0.16 mm nozzle:** 32 positions, 100 mm outer diameter, 35 mm inner diameter, 90° arc, 0.16 mm manufacturing capability. Optimiser produced fitness 1.241 with 5 tracks, 2.1 mm track width, 0.4 mm spacing, 1.8° gaps. Angular resolution: 2.8125° per position. Improved fitness reflects tighter feature packing enabled by precision nozzle.

In all cases, the optimiser identified valid configurations within 50 generations without manual parameter tuning, demonstrating adaptability across substantially different design requirements.

4 Discussion

4.1 Effectiveness of Evolutionary Optimisation

The genetic algorithm successfully navigated multi-dimensional parameter spaces subject to complex constraint interactions, but this success emerged only after careful algorithm tuning informed by initial failures. Early implementations exhibited pathological premature convergence, with identical best solutions appearing in generation 1 and persisting unchanged through all subsequent generations. The population rapidly collapsed to near-identical genomes differing only through minor mutations, eliminating the diversity necessary for continued exploration. This behaviour revealed that standard genetic algorithm configurations, whilst effective for unconstrained continuous optimisation, prove inadequate for highly constrained engineering design spaces where feasible solutions occupy relatively small volumes within the parameter space.

Fitness sharing proved critical to maintaining population diversity throughout evolution. By penalising genomes located in crowded regions of parameter space, this mechanism creates selection pressure favouring exploration of under-represented areas. The implementation computes normalised Euclidean distances between genomes in parameter space and adjusts fitness values inversely proportional to niche population density. Genomes in sparse regions receive fitness bonuses, whilst those in dense regions suffer penalties, encouraging the population to maintain diverse coverage rather than collapsing to a single peak. This transformation eliminated the premature convergence pathology entirely, enabling continued fitness improvements across dozens of generations.

Mutation parameters required substantial adjustment from conventional recommendations found in genetic algorithm literature. Initial implementations employed conservative $\pm 20\%$ perturbations with 10% mutation probability per genome. This configuration proved entirely insufficient for effective parameter space exploration in the constrained design context. Increasing mutation rate to 30–40% and magnitude to $\pm 50\%$ whilst applying mutations independently to each parameter rather than selecting a single parameter per mutation substantially improved search performance. The higher mutation rate ensures that most offspring differ from their parents, whilst the larger magnitude enables jumps across local optima that would otherwise trap the search.

Selection pressure similarly required careful calibration. Tournament selection with large tournament sizes (e.g., 5 individuals) combined with high elitism rates (10% of population) created excessive pressure toward the current best solution, reinforcing the premature convergence tendency. Reducing tournament size to 3 and elitism to 5% decreased this pressure substantially, permitting lower-fitness but potentially promising genomes to contribute genetic material to future generations. This adjustment proved synergistic with fitness sharing: reduced selection pressure creates opportunity for diverse genomes to survive, whilst fitness sharing ensures that this opportunity translates to actual diversity maintenance.

The decision to execute all generations without early stopping criteria, though computationally more expensive, proved essential for identifying high-quality solutions. Novel improvements frequently emerged in late generations (e.g., generation 47–48) after extended periods of apparent stagnation. These late improvements suggest that the algorithm was traversing relatively flat fitness landscapes where substantial parameter changes produced minimal fitness variations, but eventually discovered pathways to superior configurations. Early stopping based on convergence criteria would have terminated search

prematurely, accepting inferior local optima as final solutions.

4.2 Multi-Objective Optimisation Trade-offs

The fitness function necessarily encodes engineering judgement about the relative importance of competing objectives, and different judgements produce qualitatively different optimal configurations. The current weighting scheme allocates 40% to printability, 20% to resolution, 20% to encoding efficiency, 10% to size optimisation, and 10% to broader manufacturability concerns. This distribution prioritises manufacturing reliability over maximum resolution, reflecting the practical reality that unusable prints serve no purpose regardless of their theoretical performance. However, this represents one point in a continuous space of possible weightings, each defining a different optimisation problem with different optimal solutions.

Increasing the resolution weight to 40% whilst proportionally reducing other components drives the optimiser toward configurations with higher position counts. These designs accept reduced printability margins, approaching but not violating minimum feature size constraints, in exchange for finer angular discrimination. The resulting encoders exhibit narrower tracks and tighter spacing, increasing sensitivity to printer calibration and environmental conditions but providing superior position accuracy when successfully manufactured. This trade-off proves appropriate for applications where resolution is paramount and high-quality manufacturing capabilities are reliably available.

Conversely, increasing manufacturability weights favours conservative dimensional choices: wider tracks, larger gaps, increased spacing between features. These designs sacrifice radial space utilisation, potentially requiring larger overall diameters to accommodate the same track count, but exhibit substantial robustness to printer variations and environmental perturbations. Manufacturing success rates approach unity even with poorly-calibrated equipment or marginal material quality. This conservative approach suits applications where reliability under adverse conditions outweighs performance optimisation.

The modular architecture maintains clean separation between fitness function definition and the optimisation mechanism itself. Alternative fitness functions can be substituted without modifying genetic algorithm implementation, parameter validation logic, or geometric generation code. This separation enables systematic exploration of fitness landscape sensitivity to weighting choices, supporting informed decisions about appropriate trade-offs for specific application contexts.

4.3 Limitations and Future Work

4.3.1 Optical Sensing Model

The current system validates geometric and manufacturing constraints comprehensively but treats optical sensor performance only implicitly through minimum feature size requirements. This approach proves adequate for initial design but omits potentially significant factors affecting sensing reliability. A physics-based optical model incorporating sensor aperture geometry, alignment tolerances, light source characteristics, phototransistor sensitivity curves, and material transmission properties would enable more accurate prediction of sensing reliability across the parameter space. Such a model could account for environmental factors including ambient light interference and surface contamination effects that degrade optical contrast. Integration of an optical performance model into the

fitness function would support automatic parameter adjustment for varying sensor specifications and operating conditions, potentially identifying configurations that optimise optical signal-to-noise ratio rather than merely satisfying minimum geometric constraints.

The development of such a model faces practical challenges. Optical sensor datasheets typically provide limited information about angular sensitivity and spectral response characteristics. Material properties of 3D-printed plastics vary with printing parameters, layer orientation, and environmental exposure history. Quantitative modelling would require empirical characterisation of these dependencies, involving systematic measurement campaigns correlating geometric parameters with measured sensor performance. However, even approximate optical models incorporating first-order effects might substantially improve optimisation outcomes by capturing trade-offs invisible to purely geometric analysis.

4.3.2 Multi-Objective Pareto Frontiers

The current weighted-sum fitness function necessarily collapses multiple competing objectives into a single scalar value, implicitly defining a specific trade-off among objectives through the weight coefficients. This approach efficiently produces a single high-quality solution but obscures the structure of trade-offs available within the design space. Alternative approaches using Pareto-based multi-objective evolutionary algorithms such as NSGA-II [3] would instead identify the Pareto-optimal frontier: the set of solutions where no objective can be improved without degrading at least one other objective. This frontier reveals the range of available trade-offs, enabling designers to select configurations matching their specific priorities rather than accepting predetermined weightings.

For encoder design, the Pareto frontier would likely reveal solutions ranging from highly conservative designs optimised for manufacturing reliability with modest resolution, to aggressive designs maximising resolution whilst accepting reduced printability margins. Intermediate solutions would populate the frontier, each representing a different balance among competing objectives. Visualising this frontier supports informed decision-making by making trade-offs explicit rather than implicit in weight coefficients. The modular architecture would support Pareto-based algorithms straightforwardly by modifying the selection mechanism whilst retaining the multi-objective fitness evaluation structure.

4.3.3 Empirical Validation

Whilst the optimised design satisfies all computational validation criteria, empirical testing remains essential for confirming that simulated predictions correspond to physical reality. Planned validation activities span manufacturing verification, optical characterisation, mechanical testing, and operational evaluation. FDM fabrication using PETG material on available AnyCubic Vyper equipment will establish whether predicted feature dimensions emerge in physical artefacts or whether printing process variability introduces systematic deviations requiring model refinement. Dimensional verification via precision caliper measurements and optical microscopy will quantify actual feature sizes and their correspondence to designed values.

Optical characterisation using infrared LED and phototransistor pairs will measure signal amplitude and noise characteristics across the full angular range, identifying whether predicted optical discrimination thresholds hold under realistic sensing conditions. Mechanical testing including vibration resistance and thermal cycling will establish oper-

ational reliability envelopes. Integration testing with the ROS-2 autopilot system will validate that position data streams meet timing and accuracy requirements for effective course control. Ultimately, operational validation during actual sailing trials will provide the definitive test of whether the complete system functions reliably under authentic marine conditions.

These empirical results will necessarily inform refinement of validation criteria and fitness function weightings. Systematic discrepancies between predicted and observed performance indicate model inadequacies requiring correction. Unexpected failure modes suggest additional constraints that should enter the fitness function. The iterative refinement of computational models through empirical feedback represents standard engineering practice; the current implementation provides a foundation for this ongoing process rather than a final optimised solution.

4.3.4 Alternative Encoding Schemes

Gray code provides the single-bit-change property central to error-resistant position sensing, but alternative encoding schemes offer different trade-offs that may prove advantageous in specific contexts. Excess-3 Gray code variants employ alternative construction algorithms producing different transition patterns whilst maintaining the single-bit-change property. These alternatives might provide more balanced transition distributions across tracks or optimise for specific optical sensor characteristics. Maximum-length shift register sequences (*m*-sequences) offer pseudo-random properties potentially advantageous for noise immunity at the expense of increased decoder complexity. Absolute rotary encoders incorporating cyclic redundancy check codes enable error detection and correction, potentially identifying and recovering from transient sensing failures that would otherwise propagate as position errors.

The modular architecture specifically anticipates such explorations by maintaining clean separation between the encoding subsystem and geometric generation. Alternative encoding implementations need only satisfy the same interface—providing bit patterns for each position—enabling direct substitution without modifying geometric or optimisation components. This architectural decision reflects the understanding that encoder design represents a broader design space than Gray code implementations alone, with room for innovation in encoding theory, geometric realisation, and optimisation methodology.

4.4 Broader Applicability

The parametric design system, whilst developed specifically for marine rudder position sensing, embodies principles and techniques applicable across diverse position sensing contexts. Robotic manipulators require joint encoders spanning various angular ranges with mounting constraints determined by actuator geometry. The same parametric generation and evolutionary optimisation approach applies directly, with adjusted dimensional parameters and potentially different fitness function weightings reflecting robotic application requirements. Industrial machinery position feedback systems exhibit similar needs: absolute position knowledge, high reliability, constrained mounting geometry, but potentially different resolution requirements and environmental conditions.

The system proves particularly valuable for custom encoder design addressing legacy equipment maintenance. Proprietary encoders for older machinery may become unavailable as manufacturers discontinue product lines or exit markets entirely. The parametric

design capability enables reverse engineering of replacement encoders matching original specifications whilst incorporating modern manufacturing processes and improved reliability characteristics. This application extends equipment operational lifetime without requiring complete control system replacement.

Beyond practical applications, the system provides educational value for demonstrating Gray code properties, evolutionary optimisation principles, and multi-objective design under constraints. The accessible implementation and moderate computational requirements enable classroom deployment, whilst the concrete engineering context grounds abstract algorithmic concepts in tangible applications. The open-source release facilitates adoption across these diverse contexts without requiring substantial development effort for each new application domain.

5 Conclusions

This work demonstrates that parametric design systems coupled with evolutionary optimisation successfully address complex engineering problems characterised by competing constraints and multi-dimensional parameter spaces. The genetic algorithm implementation navigated the intricate landscape of manufacturing limitations, optical requirements, geometric constraints, and mechanical considerations to identify encoder configurations substantially superior to manually-tuned designs. The fitness improvements from initial random populations averaging 0.41–0.91 to converged solutions achieving 1.11–1.14 represent meaningful advances in practical design quality, corresponding to 11–14% improvement over baseline manual designs.

The success of this approach extends beyond the specific numerical improvements to reveal broader insights about engineering design methodology. Traditional manual design proceeds through iterative refinement guided by designer intuition about promising parameter combinations. This approach works effectively when parameter spaces are low-dimensional and constraints are well-understood, but struggles as complexity increases. The encoder design problem illustrates this limitation: the interaction between track width, spacing, gap size, and their combined effects on printability, optical performance, and geometric validity creates a constraint landscape too complex for intuitive navigation. The genetic algorithm’s identification of non-obvious parameter combinations—5.30 mm track width paired with 0.97 mm spacing and 3.23° gaps—demonstrates that automated search discovers solutions beyond the reach of conventional design intuition.

The modular software architecture proved essential to managing this complexity. Clean separation between mathematical abstractions of Gray code theory, geometric realisation through constructive solid geometry, constraint validation across multiple domains, and evolutionary optimisation enables independent development and verification of each subsystem. This separation facilitates future enhancements: physics-based optical modelling integrates cleanly into the fitness evaluation pipeline; Pareto-based multi-objective optimisation replaces the selection mechanism without affecting geometric generation; alternative encoding schemes substitute through well-defined interfaces. The architecture embodies the principle that complex systems require careful management of dependencies and abstractions to remain comprehensible and maintainable.

Adaptability emerged as a key practical advantage of the parametric approach. The system successfully generated valid encoder designs across substantially different specifications—16 to 64 positions, 80 to 120 mm diameters, 90° to 360° arcs—with manual

parameter tuning for each case. This adaptability proves particularly valuable in marine contexts where component replacement may require rapid design and fabrication of custom parts under resource constraints. The Parts Replication System aboard Sérénité, combining desktop FDM printers with parametric design tools, transforms what would traditionally require specialised manufacturing facilities into an onboard capability accessible during extended voyages.

The success of evolutionary optimisation in this domain suggests broader applicability to engineering problems sharing similar characteristics: multi-dimensional parameter spaces, interacting constraints, competing objectives, and complex design-performance relationships. The combination of domain knowledge encoded in fitness functions and validation criteria with unconstrained algorithmic exploration produces solutions balancing competing objectives in ways that may not emerge from traditional design methodologies. The fitness function and constraints represent human engineering judgement about what constitutes good design; the genetic algorithm contributes efficient exploration of consequences flowing from those judgements.

Empirical validation through fabrication and operational testing will provide the definitive assessment of whether computational optimisation translates to improved physical performance. Integration with the SCANS autopilot system during sailing trials will demonstrate whether the encoder functions reliably under authentic marine conditions or whether unforeseen factors compromise performance despite satisfying all computational validation criteria. These results will necessarily inform continued refinement of models, constraints, and optimisation strategies. The current work provides a foundation for this ongoing process rather than claiming to have solved encoder design definitively.

The open-source implementation invites community adaptation and extension. The modular architecture, comprehensive documentation, and example configurations lower barriers to adoption for diverse position sensing applications. Whether the community embraces this approach, identifies limitations requiring correction, or extends capabilities in unforeseen directions will determine the work’s ultimate impact beyond the specific marine application that motivated its development. By making the implementation freely available, we enable these possibilities whilst acknowledging that tool utility emerges through actual use rather than predicted potential.

Acknowledgements

This work was conducted independently aboard SV Sérénité. The Parts Replication System (PRS) providing onboard additive manufacturing capability was inspired by NASA’s in-space fabrication research for Mars missions. The author thanks the open-source communities maintaining Python, SolidPython, OpenSCAD, and PyQt6, without which this implementation would not have been possible.

Data Availability

Source code, documentation, and example configurations are available at:

<https://github.com/susannecoates/gray-encoder-disk-generator>

Generated encoder geometries and optimisation results are included in the repository’s `output/` directory.

References

- [1] Gray, F. (1953). Pulse code communication. U.S. Patent 2,632,058.
- [2] SolidPython contributors. SolidPython: Python interface to the OpenSCAD declarative geometry language. <https://github.com/SolidCode/SolidPython>
- [3] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2), 182–197.
- [4] Open Robotics. Robot Operating System 2 (ROS 2). <https://docs.ros.org/>
- [5] OpenSCAD contributors. OpenSCAD: The Programmers Solid 3D CAD Modeller. <https://openscad.org/>
- [6] KiCad contributors. KiCad EDA: A Cross Platform and Open Source Electronics Design Automation Suite. <https://www.kicad.org/>
- [7] Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley.
- [8] Mitchell, M. (1998). An Introduction to Genetic Algorithms. MIT Press.
- [9] Eiben, A. E., & Smith, J. E. (2015). Introduction to Evolutionary Computing (2nd ed.). Springer.