Erich Styger

Introduction

Absolute Digital Angle Encoder

Binary-Reflected Gray Code

Constructing n-B Gray Code

Converting Binary
Code into Gray Code

Converting Gray Code into Binary Code

Gray Absolute Angle

Simple Regular

Quadrature Encoder

Realtime Aspects

Quadrature

Summary

Position Encoder

Erich Styger

erich.styger@hslu.ch Lucerne University of Applied Sciences and Arts

Introduction

Erich Styger

Introduction Absolute Digital

Binary-Reflected

Constructing n-Bi Gray Code

Converting Binary Code into Gray Code

Converting Gray Code into Binary

Gray Absolute Angle

Simple Regula

Elicodei

Quadrature Encoder

. . . .

Summary

- Need to know the position of robot
- Gyroscope, accelerometer, GPS, ...
- Wheel/shaft sensors: incremental or absolute sensors.
- Will use position information for speed estimation

NTRO

Erich Styger

Introduction

Absolute Digital Angle Encoder

Binary-Reflected Gray Code

Constructing n-Bit Gray Code

Converting Binary Code into Gray Code

Code into Binary Code

Gray Absolute Angle Encoder

Simple Regu Encoder

Quadrature Encode

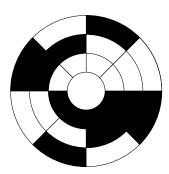
Dooltima Aspesta

Quadrature

Summary

Absolute Digital Angle Encoder

- Encoder with sectors attached to shaft/wheel
- ▶ 2ⁿ sensors for sector identification
- ► Mechanical issue: multiple bits changing



INTRO

Erich Styger

Introduction

Absolute Digital Angle Encoder

Binary-Reflected Gray Code

Gray Code

Converting Gray

Gray Absolute Angle

Simple Regul

Quadrature Encode

Realtime Asp

Quadrature

Summary

Binary-Reflected Gray Code

Frank Gray, Bell Labs, patented 1947

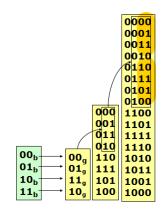
Hamming distance 1: two sucessive codes differ by one bit

Same number of bits for Gray and Binary code

Permutation: every code is present only once

Cyclic: last code rolls over to first code

▶ Recursive: G_n is embedded in G_{n+1}



NTRO

Constructing n-Bit Gray Code

- 1. Reflection: Build reverse n-1 list
- 2. Concatenate the two lists
- 3. Prefix old entries with 0
- 4. Prefix new entries with 1

Erich Styger

Introduction

Absolute Digital

Binary-Reflected Grav Code

Constructing n-Bit Gray Code

Converting Binary Code into Gray Cod

Converting Gray Code into Binary

Gray Absolute Angle Encoder

Simple Regular

Encoder

Quadrature Encode

Realtime Aspect

Quadrature Decoding

Converting Binary Code into Gray Code

- 1. Logical Shift right binary code
- 2. EXOR with binary code

$$Value_g = (Value_b >> 1) \otimes Value_b$$
 (1)

.....

Erich Styger

Introduction

Absolute Digital Angle Encoder

Binary-Reflected Gray Code

Constructing n-Bit Gray Code

Converting Binary Code into Gray Code

Converting Gray Code into Binary

Gray Absolute Angle Encoder

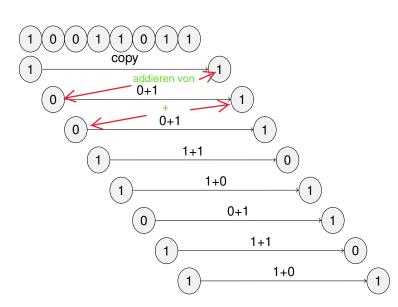
Simple Regular

Ouadratura Enacdo

Quadrature Encoder

Quadrature

Converting Gray Code into Binary Code



Erich Styger

Introduction

Absolute Digital

Binary-Reflected

Constructing n-Bi

Converting Binary

Converting Gray Code into Binary Code

Gray Absolute Angle Encoder

Simple Regula Encoder

Quadrature Encoder

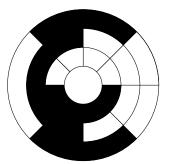
.

Quadrature

Gray Absolute Angle Encoder

- Only one bit changes
- ▶ Need 2ⁿ bits

nur immer ein bit ändert!



NTRO

Erich Styger

Introduction

Absolute Digital

Binary-Reflected

Constructing n-Bi

Converting Binary
Code into Gray Code

Converting Gray Code into Binary

Gray Absolute Angle Encoder

Simple Regular

Quadrature Encoder

Quadrature

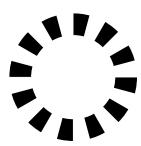
NTRO

Simple Regular Encoder

Incremental encoder

► Gray code: 0 1 0 1 ...

No backward/forward information



1141110

Erich Styger

Introduction

Absolute Digital Angle Encoder

Binary-Reflected Gray Code

Constructing n-Bi

Converting Binary Code into Gray Code

Converting Gray Code into Binary Code

Gray Absolute Angle Encoder

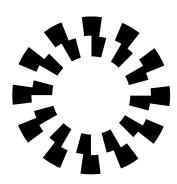
Simple Regular Encoder

Quadrature Encoder

Quadrature

Quadrature Encoder

- Incremental encoder
- 2 binary signals
- Hardware implementation: one ring with 2 LED, Sensors with offset position



Erich Styger

Introduction

Absolute Digital Angle Encoder

Binary-Reflected Gray Code

Constructing n-Bit Gray Code

Converting Binary Code into Gray Code

Converting Gray Code into Binary Code

Gray Absolute Angle Encoder

Simple Regula Encoder

Quadrature Encoder

Quadrature Encode Implementation

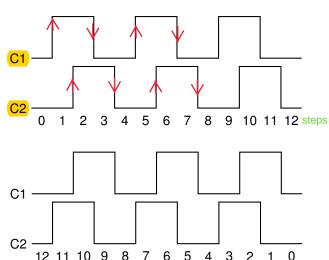
Realtime Aspects

Quadrature Decoding

Quadrature Encoder

- ► 2 binary signals, Sinus-like
- ► Shifted by 90° (one quadrant)

Problem: to many interrupts



Introduction

Absolute Digital

Binary-Reflected Gray Code

Constructing n-Bit Gray Code

Converting Binary Code into Gray Code

Code into Binary Code

Gray Absolute Angle Encoder

Simple Regula Encoder

Quadrature Encoder

Quadrature Encoder Implementation

Realtime Asp

Quadrature Decoding

Realtime Aspects

- Data acquisition
 - ► Interrupts, Input Capture
 - Sampling
 - Hardware
- Sampling: Nyquist/Shannon

$$f_{\text{sampling}} > 4 * N * 2 * R \tag{2}$$

$$t_{sampling} < \frac{\frac{1s}{4*100*2}}{\frac{4000}{60}} = 18.75 \mu s$$
 (3)

Erich Styger

Introduction

Absolute Digital Angle Encoder

Binary-Reflected Grav Code

Constructing n-Bi Gray Code

Converting Binary
Code into Gray Code

Converting Gray
Code into Binary

Gray Absolute Angle Encoder

Simple Regular Encoder

Quadrature Encoder

Realtime Aspects

Quadrature

Quadrature Decoding

- 2-dimensional array
- Previous and actual Gray Code as index
- Error correction
 - pprev, prev and new
 - In case of error, look at pprev

Erich Styger

Introduction

Absolute Digital

Binary-Reflected Grav Code

Constructing n-Bi

Converting Binary Code into Gray Code

Converting Gray Code into Binary

Gray Absolute Angle Encoder

Simple Regula Encoder

Quadrature Encode

Quadrature Decoding

Error Correc

- Build index from quadrature bits
- Interrupt context
- Error correction: additional overhead

```
void QUAD Decode(void) {
  int8 t new step:
  uint8 t c12; /* value of the two sensor input */
  it should be fast as possible! is that fast? this section isn't reentrant
  c12 = (C1\_GetVal()!=0?2:0)|(C2\_GetVal()!=0?1:0);
  new step = QUAD Table[QUAD last quadrature value][c12];
  if (new step==QUAD ERROR) { /* error case */
    QUAD errors++;
  QUAD last quadrature value = c12;
  QUAD currPos += new step;
```

reentrant problem if: in case if it's interrupted and multiple tasks using it -> make shure that only one is using that and then you have no problem

Introduction

Absolute Digital Angle Encoder

Binary-Reflected Gray Code

Constructing n-Bit Gray Code

Converting Binary Code into Gray Code

Converting Gray Code into Binar Code

Gray Absolute Angle Encoder

Simple Regulation Encoder

Quadrature Encode

Quadrature Decoding

Error Correction

Summary

- ► Absolute digital encoder: Problematic with binary code
- Binary Reflected Gray Code: Only one bit changes from code to code
- Convert Codes: EXOR and iterative bit processing
- Signal Acquisition: Hardware, Interrupt and Sampling
- Signal Processing: Table with index
- ► Realtime aspects: Error correction, interrupt latency

Erich Styger

Introduction

Absolute Digital Angle Encoder

Binary-Reflected Gray Code

Gray Code

Converting Binary
Code into Gray Code

Code into Binary

Gray Absolute Angle Encoder

Simple Regular Encoder

Quadrature Encode

Realtime Aspects

Quadrature