Design, Analyze, and Implement Radar Sensors' Alignment Algorithm with MATLAB

Liang Ma
05/13/2014
Delphi Electronically Scanning Radar

Three main properties of radar targets:
- Range
- Range rate
- Azimuth

Main features:
- Adaptive Cruise Control
- Collision Mitigation
- Rear and Side Detection System
Alignment Algorithm

- Radar sensor misalignment angle is the angle difference between real radar boresight and designed radar boresight.
- Lateral offset = range * sin(azimuth)
Least Square solver for Dynamic Alignment

Combine all the equations/detections:

\[ A(\text{azimuth}) \cdot x = B(r') \]

\[ x = A \backslash B \]
Simulation results (No noise added, injected misalignment angle: 0.3 degree)

Input host speed, radar targets’ real time azimuth, range and range rate, then simulate the LS solver alignment algorithm with MATLAB.

The simulation results shows that there is no error on the calculated misalignment angle. This algorithm is effective.
Cont. \( \text{(Add azimuth noise : } 0.3^\circ \cdot \text{randn} + 0.1^\circ \) \)

Raw Misalignment Angle Updates. Median: 0.39781 degree
Cont. (Add range-rate noise : 0.1*randn)

The greater the angle spreading, the less the noise, why?
A Matrix Condition Number ($\text{cond}(A)$)

Condition number is a relative error magnification factor.


$\text{Cond}(A)$ keeps decreasing with the increasing angle spreading of detections’ azimuth.
Real Vehicle Logs Processing Results (600+ logs, 80+ GB, 3 days' testing)
Cont.

**Raw misalignment angles**

- Graph showing misalignment angles plotted against updates number.
  - X-axis: Updates number (0 to 5.5 x 10^5).
  - Y-axis: Misalignment angle in degrees (-10 to 10).
  - Data points scattered across the graph indicating a range of misalignment angles.

**Histogram of raw angle updates**

- Graph showing a histogram of angle updates.
  - X-axis: Misalignment angle in degrees (-15 to 20).
  - Y-axis: Incident number (0 to 15000).
  - A peak indicating the frequency of updates at around 0 degrees.

---

*Delphi E&S*
Implement with Matlab Coder
MATLAB Code, C/C++ Static Library

File: alignmentAlgorithm.cpp

C source code generated on: Fri Apr 11 19:00:10 2014

#include "alignInitialization.h"
#include "alignmentAlgorithm.h"
#include "alignmentOTGvelEstimation.h"

void alignmentAlgorithm(ALIGN_OUTPUT_T *alignment_output, ...

Summary

C source code generated on: 11-Apr-2014 19:00:14

Coding target: Static Library
Number of errors: 0
Number of warnings: 0

Tell Us What You Think
We value your feedback. Please take a few minutes to answer this short questionnaire regarding the Code Generation Report.
% [YC_output, YC_Cals, YC_Params] = yawRateComp(YC_input, YC_output, YC_Cals, YC_Params);
% MATLAB Specs found here
% [YC_output, YC_Cals, YC_Params] = yawRateComp_mex(YC_input, YC_output, YC_Cals, YC_Params); % Verify generated code here and accelerate
Coder Code Performance

Conclusions:

✓ Reliable. Coder code has been used in production code for half a year and no bug is found.
✓ Efficient. This improved alignment algorithm with coder code can run as fast as previous old algorithm with hand code.
✓ Easy to integrate.
Summary

- Radar Sensors’ Alignment Algorithm design, simulation, data analysis, and implementation are done together within Matlab only;
- One algorithm engineer can completely responsible with one algorithm block. Save time to coordinate with software engineer.
- In the future, Matlab plus Coder will be popular in algorithm development area.
DELPHI
Innovation for the Real World