Technical Simulation Interview

These small technical problems represent some of the problems that a member of the simulation team needs to be able to solve. No specific technology is required for the solution. Block diagrams drawn on paper or pseudocode could do! However, we will positively value any implemented resolution that is ready to execute (for example, compilable software, or a Simulink model).

Wheelspeed virtual model

The goal of this activity is to create a virtual model that represents the sensors that measure the rotational speed of the wheel. The model does not need to be perfectly accurate to the real world, but it needs to be functionally equivalent. If required, it could be refined at a later stage in development for a more realistic behavior, but this is not a goal of this activity

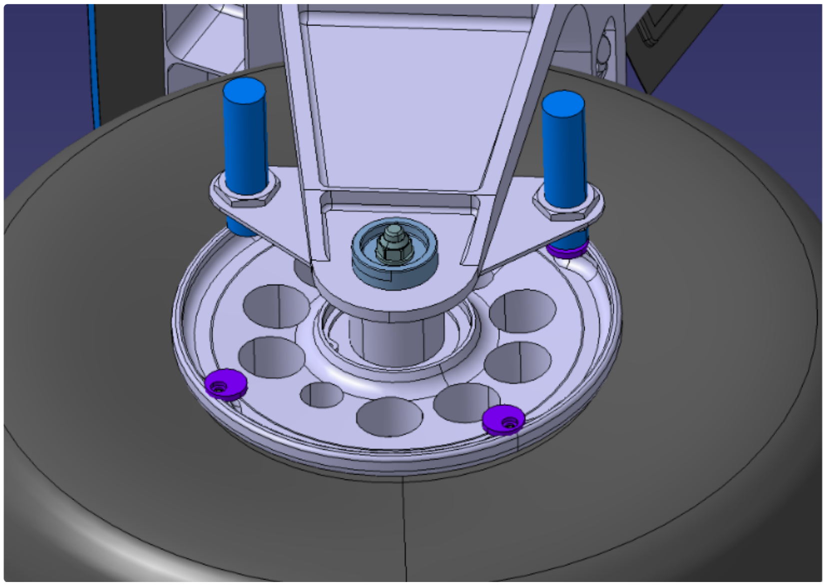
Sensor

The sensors themeselves are proximity sensors. When a ferrous material comes close to the sensing end, the signal pin goes high by means of a small transistor inside the sensor body; otherwise it stays low. In addition to the signal pin, the sensor has a GND and a power supply pin (these sensors operate at 28V).

Installation

For redundancy, there are two identical sensors mounted to one side of the Main Landing Gear wheel. The wheel itself has 4 ferromagnetic bolts that rotate with the rim and each of them come close enough to each of the sensors once every revolution of the wheel

The diameter of wheel is 6”



Model context

The dynamics of the aircraft are provided by a separate physics model. You can consider the general state of the aircraft to be an input to this model. In particular, some of the signals that are generated by the physics model are given below. You need not use all of them for this model, but some will be useful.

- ins
  - geo_position
    - latitude
    - longitude
    - altitude
  - attitude_euler
    - roll
    - pitch
    - yaw
  - ground_velocity
    - vx
    - vy
    - vz
  - body_angular_rates
    - p
    - q
    - r
  - body_acceleration
    - ax
    - ay
    - az
  - ground_acceleration
    - ax
    - ay
    - az
  - wind_velocity
    - vx
    - vy
    - vz

- weight_on_gear
  - proximity_sensor

All the INS observable signals are represented as double precision floating point numbers (float64), given in SI of units. The weight_on_gear.proximity_sensor is a boolean signal that represents whether the wheel is currently touching the ground or not.

Objective

The goal of this exercise is to create a model that can be used to simulate both wheelspeed sensors. The inputs are the physics observables listed above, and the outputs should be the signals that sensors can output on their signal pins. What will the signals look like when the wheel is moving? Provide some plots showing the expected waveform.

Wheel dynamics and the sensor's performance could be considered to make this model more realistic, but this is not important at this stage. Consider providing some parameters that could quantitatively affect the performance of the model with some rough initial values. These can be refined later.

It is encouraged that this model is created using Simulink, but plain source code or even logic block diagrams are perfectly acceptable.

Extra credit

You can simulate power on and power off behavior with another input signal that represents the voltage supplied to the power pin of the sensors. How would this affect the output signals?

Think of possible ways that the system might fail in real life. These modes of failure could be an additional input to the virtual model, used for failure detection by other systems downstream. List these failures and provide a small description of how it could affect the model, but do not implement them.

ARINC 429 encode

The INS provides data using a standard protocol called Arinc 429. Plenty of information on the protocol exists online. A particularly good reference is this one https://www.aim-online.com/wp-content/uploads/2019/07/aim-tutorial-oview429-190712-u.pdf but you can use any other.

Context

A small part of the INS virtual model, after doing all the internal logic, is to encode the outputs using this protocol such that it can be received by other equipment on the aircraft.

The INS model is also fed by the physics INS observables above. Use any of the signals to build your solution. Remember they are all provided in SI units.

Useful data

The INS manufacturer provides a extensive Installation Manual that provides information on how to use their equipment. A small portion of the manual will be reproduced here with the information you need for this task

- Assume you are modelling an INS that is configured with identifier 1 for the SDI portion of the label
- The SSM portion represents the validity of the label. Assume we are modelling only two cases: SSM = NCD for when the INS is still alligning, and SSM = NO after that. Assume the state st_ALIGNING (boolean) is available to your encoding function.
- The Ground Speed Label 312 data portion is encoded in this way

4.3.2.4 Digital output, ARINC 429, IRS navigation output data definition

Operating in IRS navigation mode the system provides IRS navigation output data with the label definitions as defined in the following table:

Label (octal)	Parameter	Notes	Format	Sig. Bits	Max. range	Approx. resolution	Selftest value	Units	Pos. sense
147	Magnetic variation	(5)	BNR	18	±180	0.00068	10	deg	CW from north
310	Present position latitude		BNR	20	±180	0.000172	N 22.5	deg	North
311	Present position longitude		BNR	20	±180	0.000172	E 22.5	deg	East
312	Ground speed		BNR	18	4,096	0.015	200	kts	Always positive

- The parity bit is set such that the overall label parity is always ODD

Objective

Build a function (or set of functions) that encodes the ARINC429 the label 312, which represents Ground speed data out of the INS.

Use the function to encode some example values of Ground speed (e.g. 0 m/s, 15 m/s, 4096 kts…)

It is encouraged that you provide C or Matlab source code for this task, but any other programming language is acceptable. Diagrams, tables, matematic descriptions or any other tool is okay to use.

Extra credit

From section 4.3.2.4 of the manual, you can see that the data portion of label 310 (Inertial latitude) has 20 significant bits instead of the usual 18. The SDI bits are used for this additional resolution. Make sure the function you build is generic enough that it can deal with both possibilities.

The Hybrid latitude and longitude data needs more precision than the bits available in a single Arinc label. For this reason, it is split into a Coarse and Fine labels, which can be then combined by the decoding equipment. Describe a strategy (but do not implement) for how you would split the latitude value into Coarse and Fine portions before applying your encoding function to build the labels.

Label (octal)	Parameter	Format	Max. delay [ms]	Update rate [Hz]	Sign. bits	Maximum range	Approx. resolution	Units	Positive sense
074	UTC Measure time	BNR	200	1	20	10.0	9.536743 µs	s	Always positive
076 (2)	GNSS Altitude (MSL)	BNR	200	1	20	±131,072	0.125	feet	Up
110	GNSS Latitude	BNR	200	1	20	±180	0.000172	deg	North
111	GNSS Longitude	BNR	200	1	20	±180	0.000172	deg	East
120	GNSS Latitude fine	BNR	200	1	11	0.000172	8.38E-8	deg	N/A (1)
121	GNSS Longitude fine	BNR	200	1	11	0.000172	8.38E-8	deg	N/A (1)

1) Fine data words contain the truncated portion of the original data word. This information is unsigned although the sign bit is reserved. The two data words are concatenated (or combined).