BIBLIOGRAPHY

[1] Analysis and Design of GPS based Target tracking system and MIL-STD-1553B Radar Target data
Ramakanthkumar, P.; Prabhushankar, G.; Ilavarasu, S
Advanced Computing and Communications, 2006. ADCOM 2006. International Conference
Volume, Issue, 20-23 Dec. 2006 Page(s):632 – 633

[2] Greenspan, R. L, and Donna, J. I., *Measurement Errors in GPS Observables* Journal of The Institute of Navigation, VoL 33, No. 4, Winter 1986-87

[3] Barrett, Dick, "All you ever wanted to know about British air defence radar". The Radar Pages. (History and details of various British radar systems)

[4] USAF Long Range Radar, "*84th Radar Evaluation Squadron*". US Air Force Squadron responsible for long-range Radar Sensor (both civilian and military) operational availability, counterdrug, search and rescue, and flight safety information assurance to the operations community.

[5] Hollmann, Martin, "Radar Family Tree". Radar World.

[6] A Kinematic Carrier Phase Tracking System for High Precision Trajectory Determination
Dr. Wang Tang, Mr. Dale Turley & Mr. Gene Howell
ARINC *Research Corporation*
Ms. Linda Wilkinson *Hughes Aircrafr Company*

[7] R. Michelson, "Rules for the Current International Aerial Robotics Competition Mission," http://avdil.gtri.gatech.edu/AUVS/CurrentIARC/200xCollegiateRules.html, May 2004.

[8] B.E. Tweddle et al., "A Network-Based Implementation of an Aerial Robotic System," Proceedings of the AUVS International Symposium, 2004.

[9]. *MPX5010 Integrated Silicon Pressure Sensor* [online]. Freescale Semiconductor, Available from: http://www.freescale.com/files/sensors/doc/data_sheet/MPX5010.pdf

[10]. *MPX4115 Integrated Silicon Pressure Sensor* [online]. Freescale Semiconductor, Available from: http://www.freescale.com/files/sensors/doc/data_sheet/MPX4115A.pdf

[11]. *MA 4805A/B Cougar Data Interface Unit* - Technical Manual

[12] " Automatic Dependent Surveillance-Broadcast (ADS-B)". Posted on AOPA online Tuesday, January 03, 2006 2:28:54 PM

[13] "Automatic Dependent Surveillance Broadcast (ADS-B)". Federal Aviation Administration. http://www.faa.gov/airports_airtraffic/technology/ads-b/

[14] "An Introduction to the Kalman Filter" by Gary Bishop

Appendix 1 – Altitude & Pressure Relationship

Features of the earth's atmosphere at various altitudes are described by a basic set of equations under the International Standard Atmosphere (ISA) model. The atmospheric layer at 0 to 11 km is called the troposphere. This is a non-isothermal region where the temperature changes from 15^0C to -56.5^0C.

The relationship between altitude h and atmospheric pressure p is given by the following formula. Here p1 is the starting pressure of the troposphere.

$$h = \frac{[1 - \left(\frac{p}{p1}\right)^{0.19}]}{22.558 \times 10^{-6}} \text{ m} \tag{1.1}$$

This formula is valid under standard atmospheric conditions i.e. when p1 is 101,325 Pa. To account for non standard conditions and to calibrate the formula a practical p1 value can be found by the following formula.

$$p_1 = p\left[\frac{288.15 - 0.0065 \times h}{288.15}\right]^{5.256} \tag{1.2}$$

Here the current pressure reading p and the known altitude h is used to compute a compensated value for p1.

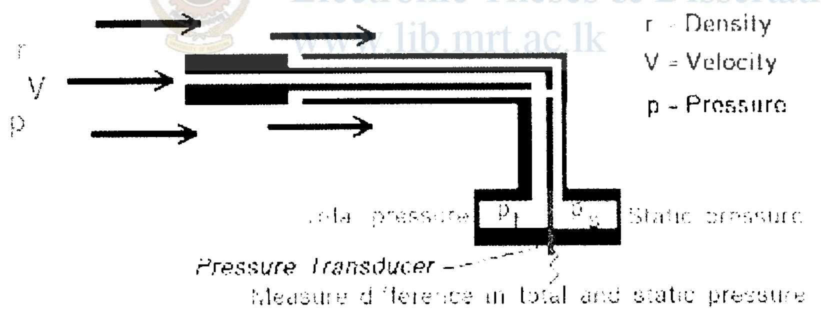
Appendix 2 – Air Speed & Pressure Relationship

Air speed is the speed of an airborne object relative to the medium of air. This definition is what is called True Airspeed (TAS). But this is almost never what airspeed indicators on airplanes actually indicate. Instead they use Indicated Airspeed (IAS).

IAS is calculated according to the following formula derived from the Bernoulli's equation. Here ΔP is called dynamic pressure which is given in Pascal.

$$\text{IAS} = \sqrt{\frac{2}{1.225}(\Delta P)}\ \ \text{ms}^{-1} \tag{2.1}$$

Dynamic pressure is the difference between the total pressure due to wind, and the static or atmospheric pressure. A Pitot tube mechanism is commonly used to feed this dynamic pressure to a pressure transducer. This is illustrated in the figure below.



Pitot Tube Mechanism

Appendix 3 – LTC 1865

Features:

- 16-Bit 250ksps ADCs in MSOP Package
- Single 5V Supply
- Low Supply Current: 850mA (Typ)
- Auto Shutdown Reduces Supply Current to 2mA at 1ksps
- True Differential Inputs
- 1-Channel (LTC1864) or 2-Channel (LTC1865) Versions
- SPI/MICROWIRETM Compatible Serial I/O
- 16-Bit Upgrade to 12-Bit LTC1286/LTC1298
- Pin Compatible with 12-Bit LTC1860/LTC1861

Appendix 4 – PIC 16F877A Microcontroller

Features:

High-Performance RISC CPU:

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program branches, which are two-cycle
- Operating speed: DC – 20 MHz clock input DC – 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory, Up to 368 x 8 bytes of Data Memory (RAM), Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to other 28-pin or 40/44-pin PIC16CXXX and PIC16FXXX microcontrollers

Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during Sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
    - Capture is 16-bit, max. resolution is 12.5 ns
    - Compare is 16-bit, max. resolution is 200 ns
    - PWM max. resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I2C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) – 8 bits wide with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (VREF) module
  - Programmable input multiplexing from device inputs and internal voltage reference
  - Comparator outputs are externally accessible

Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming (ICSP) via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

CMOS Technology:

- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

Appendix 5 – TS 7300 SBC

Features

- 200MHz ARM9 CPU
- PC/104 expansion bus
- 32MB SDRAM (64-128MB opt)
- User-programmable CycloneII FPGA
- 2 10/100 ethernet ports
- 2 USB 2.0 (12Mbit/s max)
- 2 SD Card slots
- 10 RS232 ports (more if TTL only)
- 55 DIO lines (up to 35 TS-XDIO)
- 1 VGA out w/ 8MB RAM 800x600
- Matrix Keypad and text LCD support
- Optional Temp Sensor, RTC and WiFi
- Low-power 1.8W @ 5V (CPU/SDRAM full speed, ethernets on/unplugged, all serial ports on, default FPGA bitstream)
- Fanless -40° to +70°C, +85°C 166Mhz
- TS-ENC730 Metal Enclosure provides standard 4 ADC, 2 DAC, 4 high-current GPIO, optional 1 CAN bus, etc.
- Fast boot to Linux shell in 1.69 secs
- Debian Linux is default on SD Card
- SD Card High-Security Features
- Linux-based Bootloader available
- Linux utility for FPGA configuration

Appendix 6 – Garmin GPS 18 (5Hz)

Features

- GPS performance
- Receiver: WAAS enabled; 12 parallel channel GPS receiver continuously tracks and uses up to 12 satellites to compute and update your position
- Acquisition times
- Reacquisition: Less than 2 seconds
- Warm: Approximately 15 seconds
- Cold: Approximately 45 seconds
- Auto Locate™: 5 minutes
- Sky Search: 5 minutes
- Update rate: 1 to 900 seconds between updates;
    - programmable in 1 second increments
- GPS accuracy:
- Position: < 15 meters, 95% typical**
- Velocity: 0.1 knot RMS steady state
- Dynamics: 999 knots, 6g's

Appendix 6 –Kalman Filter

Within the significant toolbox of mathematical tools that can be used for stochastic estimation from noisy sensor measurements, one of the most well-known and often-used tools is what is known as the *Kalman filter*. The Kalman filter is essentially a set of mathematical equations that implement a predictor-corrector type estimator that is *optimal* in the sense that it minimizes the estimated *error* covariance—when some presumed conditions are met. Since the time of its introduction, the *Kalman filter* has been the subject of extensive research and application, particularly in the area of autonomous or assisted navigation. This is likely due in large part to advances in digital computing that made the use of the filter practical, but also to the relative simplicity and robust nature of the filter itself [14].

The Discrete Kalman Filter

The filter in its original formulation (Kalman 1960) where the measurements occur and the state is estimated at discrete points in time.The Kalman filter addresses the general problem of trying to estimate the state $x \in \Re^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \quad (1)$$

with a measurement $z \in \Re^m$ that is

$$z_k = Hx_k + v_k \quad (2)$$

The random variables and represent the process and measurement noise (respectively). They are assumed to be independent (of each other), white, and with normal probability distributions [14].

$$p(w) \sim N(0, Q) \quad (3)$$

$$p(v) \sim N(0, R) \quad (4)$$

In practice, the *process noise covariance* Q and *measurement noise covariance* matrices might change with each time step or measurement, however here we assume they are constant. The $n \times n$ matrix A in the difference equation equation (1) relates the state at the previous time step $k-1$ to the state at the current step k, in the absence of either a driving function or process noise. Note that in practice A might change with each time step, but here we assume it is constant. The $n \times l$ matrix *B* relates the optional control input $u \in \Re^l$ to the state *x*. The $m \times n$ matrix H in the measurement equation equation (2) relates the state to the measurement *zk*. In practice H might change with each time step or measurement, but here we assume it is constant.

The Discrete Kalman Filter Algorithm

The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements. As such, the equations for the Kalman filter fall into two groups: *time update* equations and *measurement update* equations. The time update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the *a priori* estimates for the next time step. The measurement update equations are responsible for the feedback—i.e. for incorporating a new measurement into the *a priori* estimate to obtain an improved *a posteriori* estimate. The time update equations can also be thought of as *predictor* equations, while the measurement update equations can be thought of as *corrector* equations. Indeed the final estimation algorithm resembles that of a *predictor-corrector* algorithm for solving numerical problems as shown below in Figure.

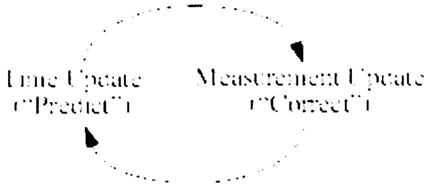


Figure 1: The ongoing discrete Kalman filter cycle. The *time update* projects the current state estimate ahead in time. The *measurement update* adjusts the projected estimate by an actual measurement at that time [14].

The specific equations for the time and measurement updates are presented below for Discrete Kalman filter time update equations

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \tag{5}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{6}$$

And Discrete Kalman filter measurement updates equations

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{7}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{8}$$

$$P_k = (I - K_k H)P_k^- \tag{9}$$

The first task during the measurement update is to compute the Kalman gain K_k by equation (9), The next step is to actually measure the process to obtain z_k, and then to generate an *a posteriori* state estimate by incorporating the measurement as in equation (8). The final step is to obtain an *a posteriori* error covariance estimate via equation (9). Figure 2 below offers a complete picture of the operation of the filter, combining the high-level diagram of Figure 4.1 with the equations presented above.

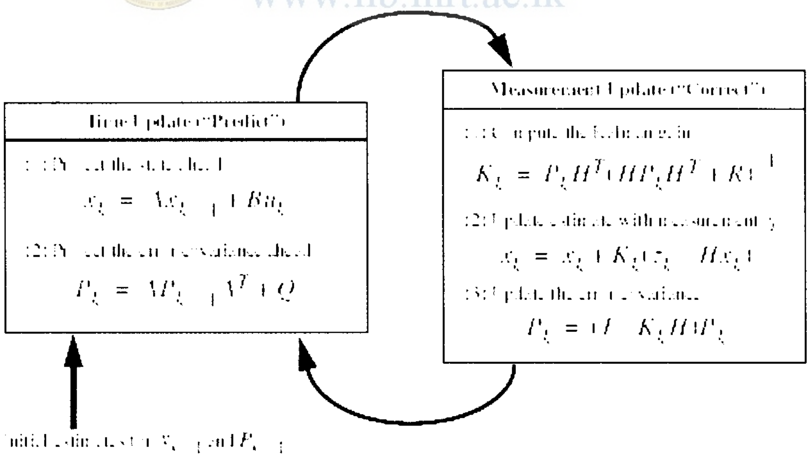


Figure 2: A complete picture of the operation of the Kalman filter, combining the high-level diagram of Figure 1 with the equations from (5) to (9). [14]