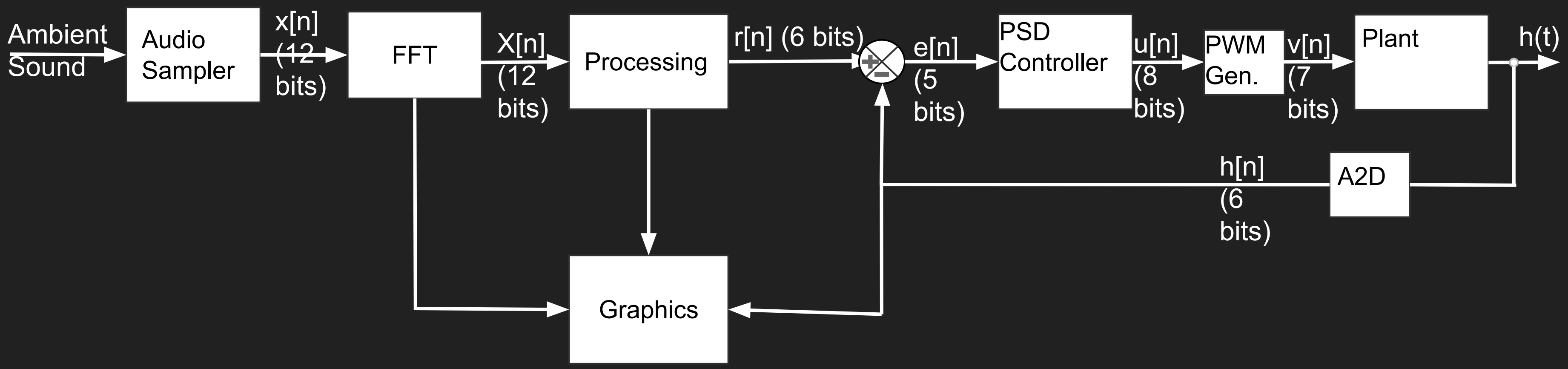
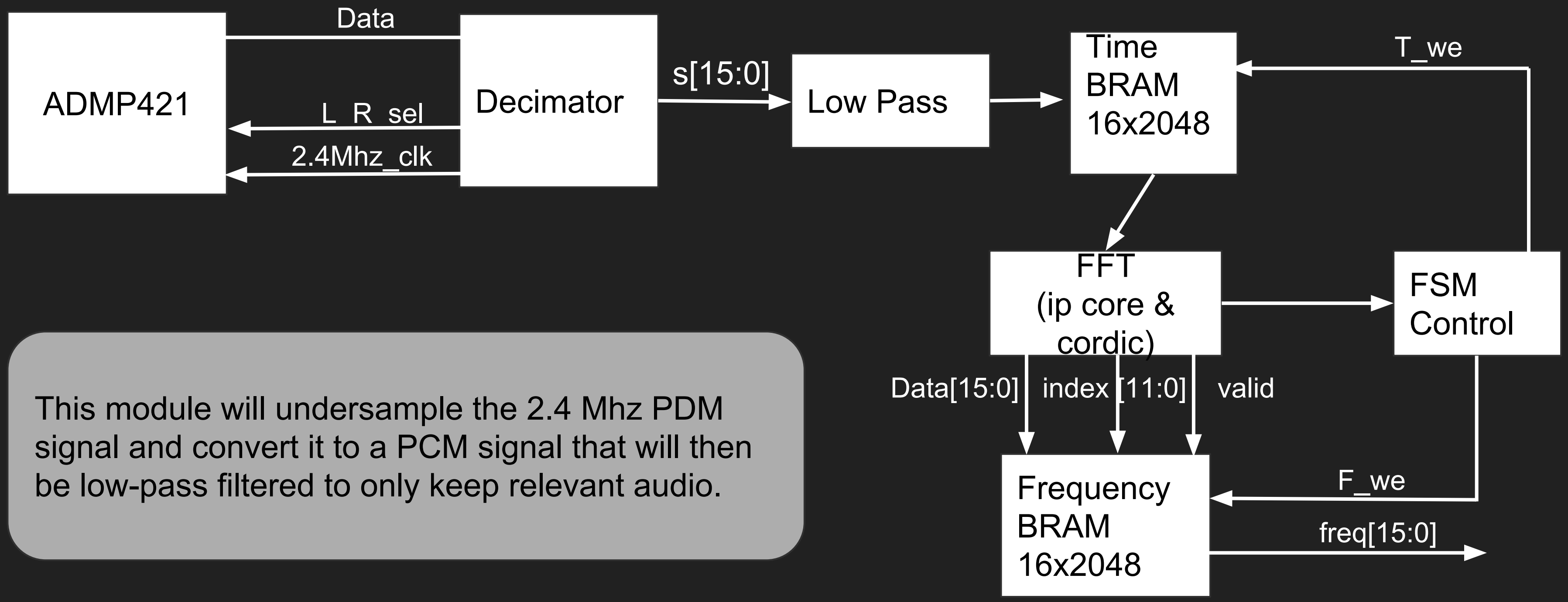
# Audio-Controlled Levitator

David Mejorado and Raul Largaespada

# High Level Block Diagram

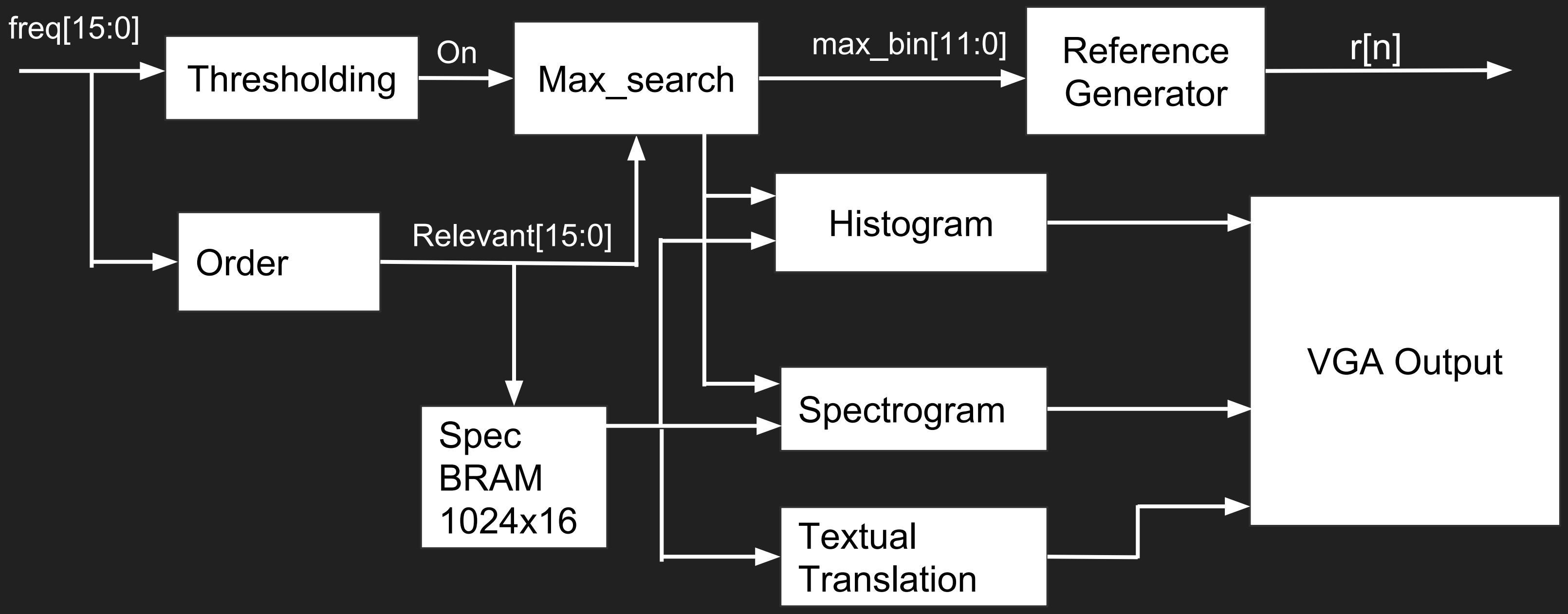


# Audio Sampler & FFT

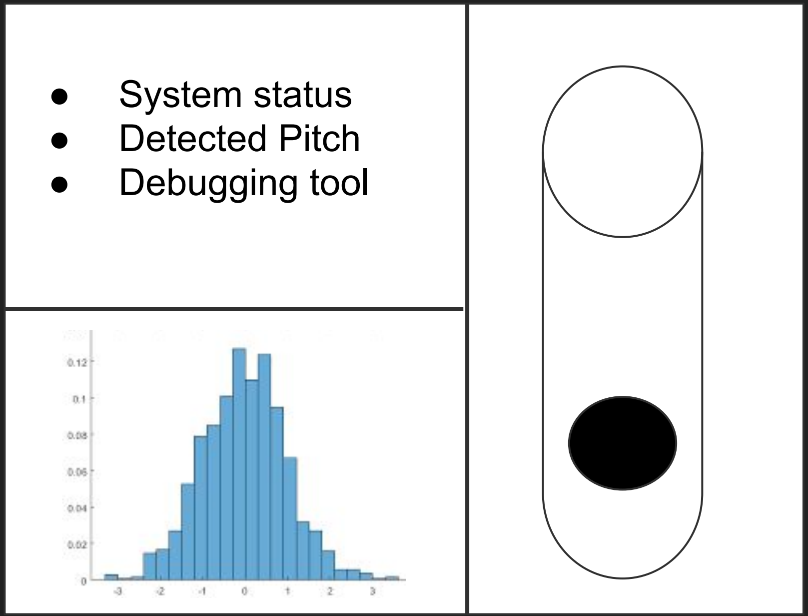



This module will undersample the 2.4 Mhz PDM signal and convert it to a PCM signal that will then be low-pass filtered to only keep relevant audio.

# Processing & Graphics



# More Details




## Base Goals

- Extract pitch
- Visualize Histogram & text description

## Stretch Goals

- Visualize spectrogram
- Animate search
- Animate expected system behavior

# Possible Complications

- Audio signal is still too noisy to distinguish dominant frequency
- Throughput of reference signal is too slow
- Amount of memory available

# System Dynamics

$$x[n] = x[n-1] + \Delta t \cdot (v[n-1])$$

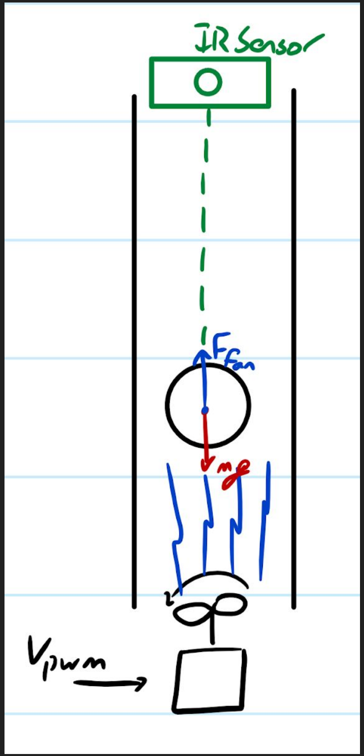
$$v[n] = v[n-1] + \Delta t \cdot (a[n-1])$$

$$a[n] = \frac{\mathrm{F_{fan}}}{\mathrm{m}} - g$$

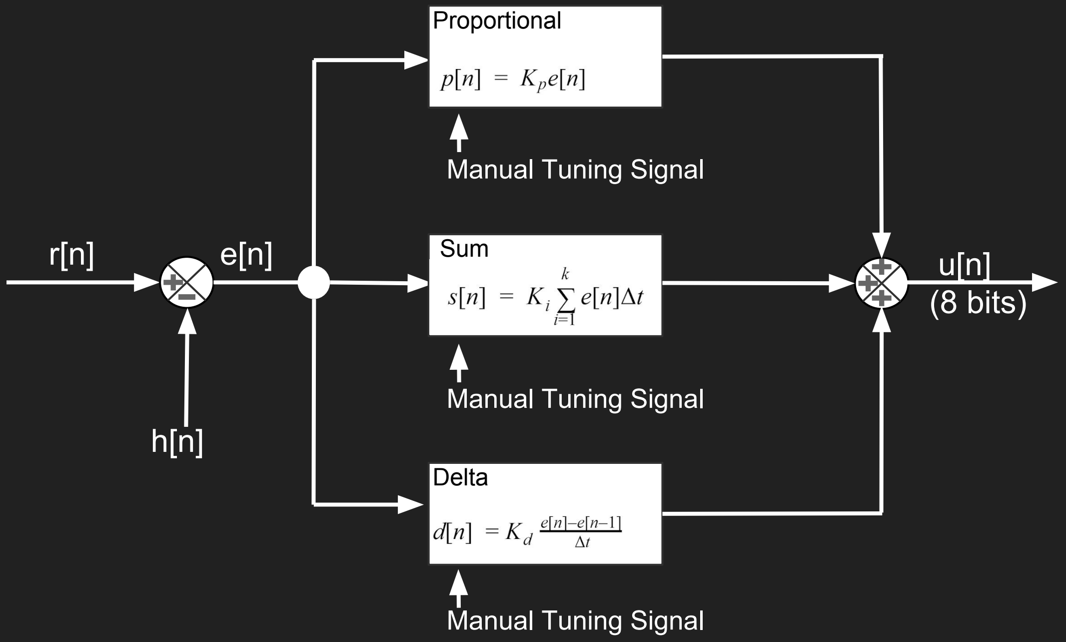
$$\mathrm{F_{fan}} = \gamma \cdot u[n]$$

$$x[n] = x[n-1] + \Delta t \cdot (v[n-1]) + \Delta \mathrm{t}^2 \cdot \left(\frac{\gamma \cdot u[n]}{\mathrm{m}} - g\right)$$

- PWM voltage from FPGA accelerates a motorized fan
- Air from fan balances ball at a specific height
- IR sensor detects height of ball for feedback control



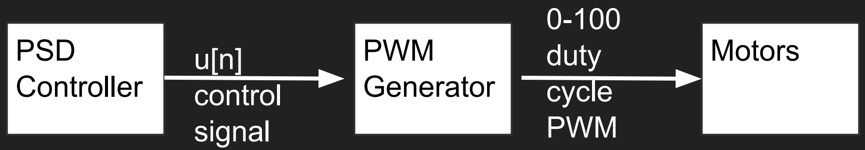
# Proportional Sum Delta Controller




- Finite state machine based PSD controller
- Gain values are hard-coded into Verilog, additional tuning supported through potentiometers
- After receiving new sensor input, will run through all computational steps before sending out new control output
- Many clock cycles to work with because sensors are slow compared to FPGA; efficiency not key

# PWM Generator, A2D

- Using an MCP3008 for A2D hardware, will only consider the 6 MSB
- Communicate using SPI, scale value to height of ball/potentiometer angle
- PWM generator will convert control signal u[n] into a 7 bit 0-100 duty cycle value

# Further Possibilities: Eigenvalue Placement/LQR

- If the PSD control architecture is successful, we would like to try additional control techniques
- Convert system dynamics to state-space form
- Attempt eigenvalue placement, linear quadratic regulator controllers calculated in MATLAB
- Will require new Verilog modules, can retain most previous hardware

$$\dot{x} = Ax + Bu$$

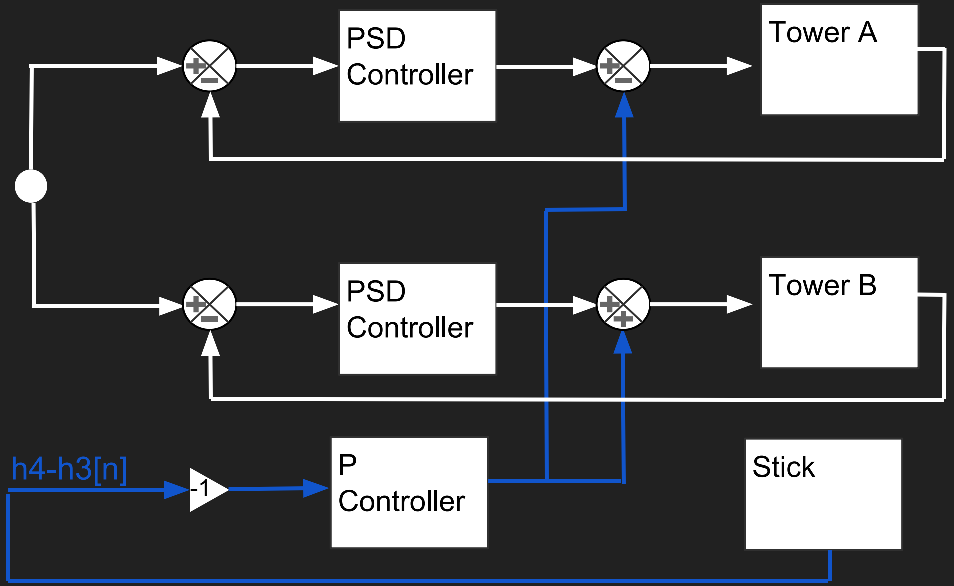
$$y = Cx + Du$$

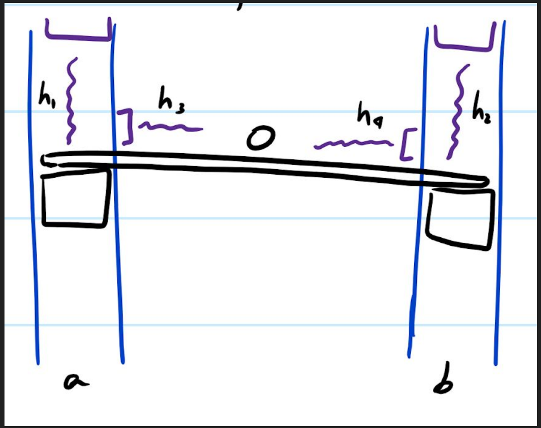
where

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{bmatrix}$$

$$J = \int_0^\infty \mathbf{x}^T Q \mathbf{x} + \mathbf{u}^T R \mathbf{u} \, dt$$

# Further Possibilities: The Two Towers



- Goal: Balancing a ball between two towers
- Restructure system to make use of three separate controllers, additional hardware

# Timeline

| Date | Raul | David |
| --- | --- | --- |
| 11/04 - 11/10 | System Modeling, Circuitry Design | FFT module and Histogram |
| 11/11 - 11/17 | PSD Implementation | Thresholding and Max Search |
| 11/18 - 11/24 | Single Tower Hardware Implementation | Integration and Text graphics |
| 11/25 - 12/01 | Stretch Goals: State-Space | Stretch Goals: More Graphics & debug |
| 12/02 - 12/09 | Stretch Goals: Two-Towers | Stretch Goals: More Graphics & debug |