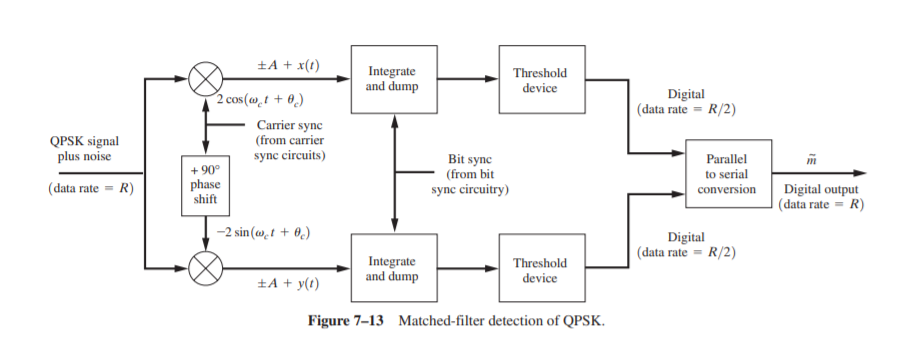
Digital design document P&D Jonas

# Coherent or Incoherent?

Since a coherent demodulator requires a reference signal to demodulate the modulated signal, which we would have to extract from the modulated signal using extra hardware. Instead of doing this, we will use a carrier sync to get as close as possible to the carrier frequency, and then use non-coherent detection (i.e. some LPF like an integrate and dump (update: actually, use a matched filter that matches the pulse shape of the MSK, which is a cos I believe, what is it in GMSK?), then a threshold device, the detection pictured is for QPSK but as pointed out in Couch, MSK is just QPSK with cos and sin pulse instead of rectangular pulses). See section 7-5 of Couch’s Digital and Analog Communication Systems.



# Automatic Gain Control:

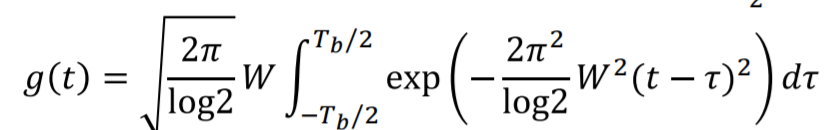
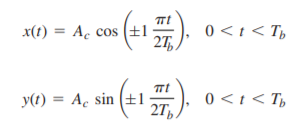
Handled by the analog development team

# Multiplication with Cos/Sin functions as step one of retrieving the in-phase and quadrature components:

Implement using CORDIC, because it is known to be well adapted to use on an FPGA

# Low Pass filter

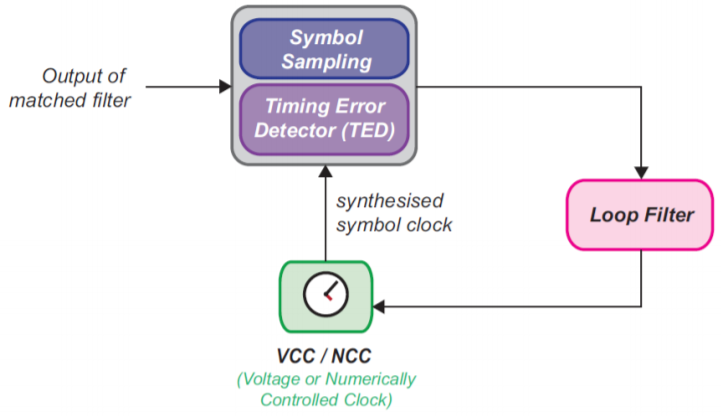
This requires some thought: What are the requirements for this low pass filter?

* How much does the magnitude response need to be depressed at certain frequencies? (Test this in matlab)
* Does the phase response of the filter need to be linear? (Yes!!!, because the phase is the information we’re looking for, and then we’d have to correct, so no IIR filters)
* The GMSK bible mentions CIC filters, (NOPE YOU CANNOT MAKE ANY FIR FILTER WITH THAT ONLY SPECIFIC ONES)
* Bandwidth: 200Hz
* Lattice Ladder realization
* GMSK pulse shape out of filter: 
* 

For non-coherent detection, we probably want a matched filter to get optimal SNR.

# Clock Recovery

Timing synchronization blocks consisting of :



TED algorithms: • Early-Late Gate • Zero Crossing (Gardner Loop) • Maximum likelihood • Mueller and Meuller.

We Choose early-Late Gate algorithms, since it is easier to implement on hardware and is widely used in digital communications.

(clock recovery can be done based on the zero crossings of the relative frequency signal?)

# Carrier Recovery

If we attempt to do coherent detection, we must do carrier synchronization. This is described by Couch on page 549 where he states

*“For QPSK signals, the carrier reference may be obtained by a more generalized*

*Costas loop or, equivalently, by a fourth-power loop [Spilker, 1977]. These loops have a fourphase ambiguity of 0, ;90, or 180°, which must also be resolved to obtain correctly demodulated data. These facts illustrate once again why noncoherent reception techniques (which can*

*be used for OOK, FSK, and DPSK) are so popular.”*

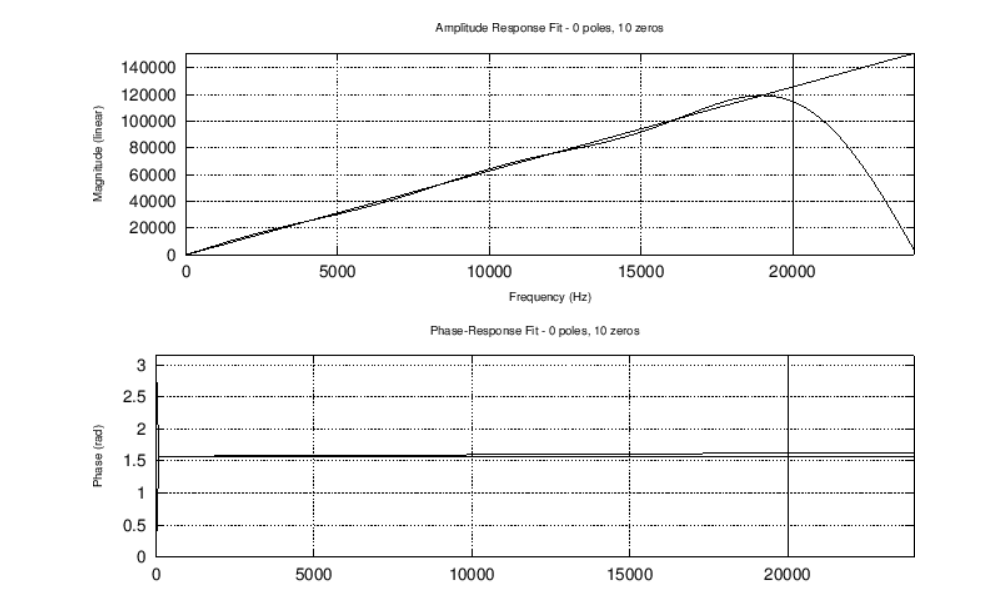
# Phase extraction

Again done via CORDIC

# Derivator

To make a derivator in the time domain, we must simply apply a filter with a linear magnitude response and a flat phase response:





We can use the same method to do this as we used to implement our LPF/Matched filter

# Convolutional code decoder

This is optional. We are trying to keep the other parts of our design as simple as possible, so we can implement this, but getting the other parts to work will be our primary concern.

# Varicode Decoder

Implementing this in matlab means using a table lookup to translate the varicode to our message.