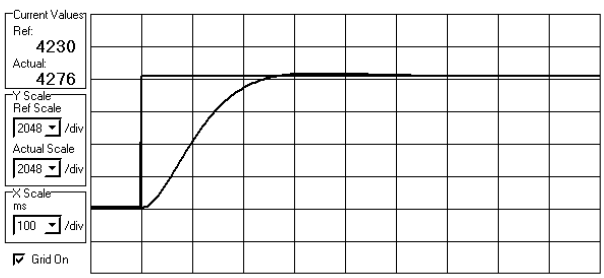
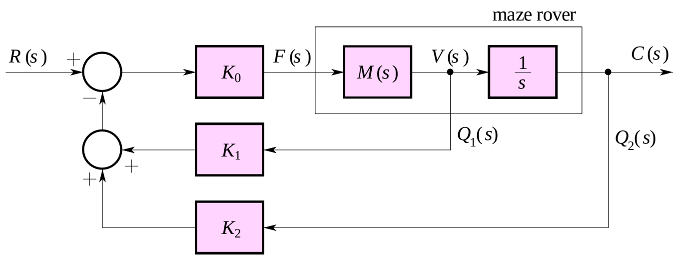
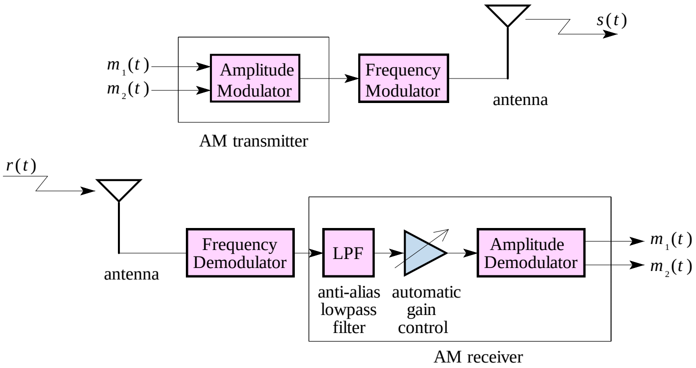
# 48540 Signals and Systems

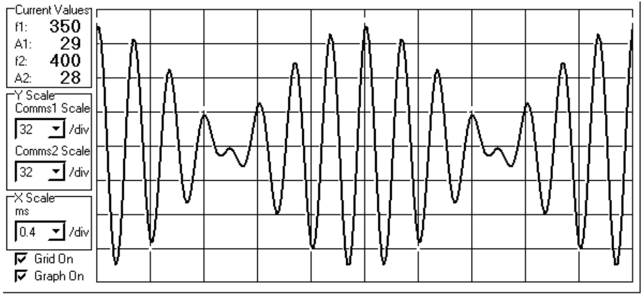
## *Project Notes*

**Spring 2015**




PMcL

# Group Project – Digital Controller

*Maze rover control system. Specifications.*

## Introduction

The MR uses a control system to respond to position commands received from its communication system. You will be designing elements of both systems. They are implemented in the μC of the MR, so they are inherently discrete-time systems. This document outlines the control system specifications. The communication system is specified in a separate document.

## Control System Overview

The control scheme is used to achieve a desired position. The MR has internal position sensing, so a feedback control system can be easily designed based on a reference position input:

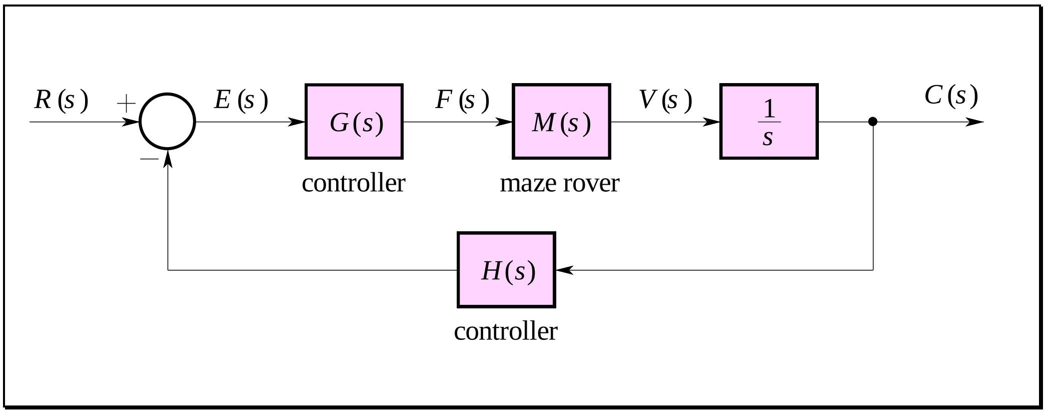



**Figure 1 - General Control Scheme**

Notice how the position is obtained from the velocity by an integrator and that a continuous-time model of the system has been used, when in fact most of the system is discrete-time. It will be easier to design the controllers using continuous-time techniques and then discretize them, rather than designing directly in the *z*-domain.

In Lab 6 you should find a continuous-time model for the open-loop MR (a transfer function) in terms of the input force $F(s)$ and the output velocity $V(s)$.

# Control.2

You now have to design the controller to meet certain time-domain performance specifications.

## Specifications

All MRs must meet the following specifications:

- The sample period is 4 ms.

Individual MRs must meet the following specifications for a "unit" step. A "unit" step for the MR will physically mean a 0 V to +2 V step.

| Group Project | 5% Settling Time (ms) | Peak Time (ms) | Percent Overshoot (%) |
|---|---|---|---|
| A | 230 | 200 | 4.0 |
| B | 220 | 190 | 4.5 |
| C | 210 | 200 | 5.0 |
| D | 240 | 180 | 5.5 |
| E | 230 | 200 | 6.0 |
| F | 210 | 220 | 6.5 |
| G | 240 | 200 | 7.0 |
| H | 270 | 180 | 7.5 |
| I | 280 | 250 | 7.0 |
| J | 250 | 210 | 6.5 |
| K | 260 | 290 | 6.0 |
| L | 230 | 220 | 5.5 |

**Note**: Lab 6 measured velocity in terms of "distance per sample period". We need the velocity in terms of "distance per second". You therefore need to multiply the transfer function of the MR obtained in Lab 6 by a scaling factor to correct for this.

## Control Schemes

The following three control schemes are being trialled. You will need to design controller coefficients for each of them.

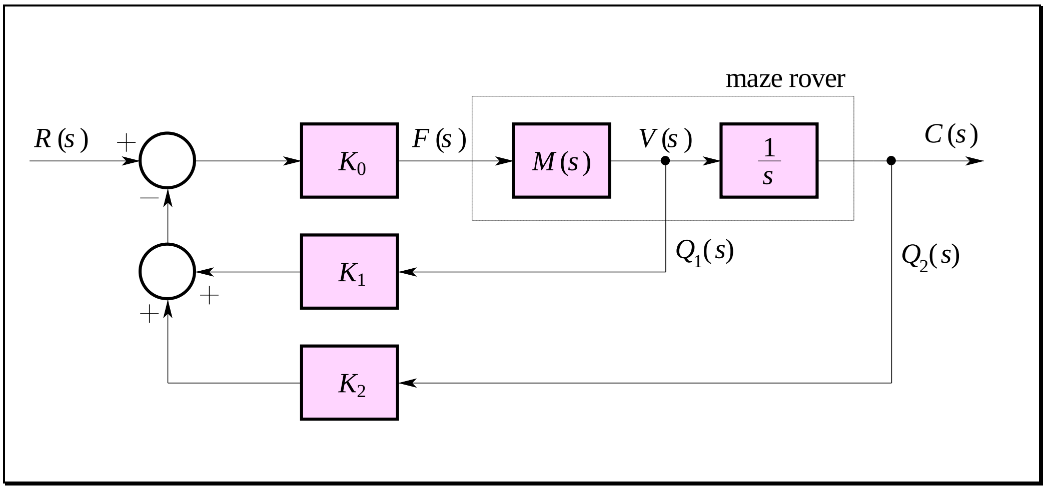



**Figure 2 - Scheme 1 – Linear State-Variable Feedback**

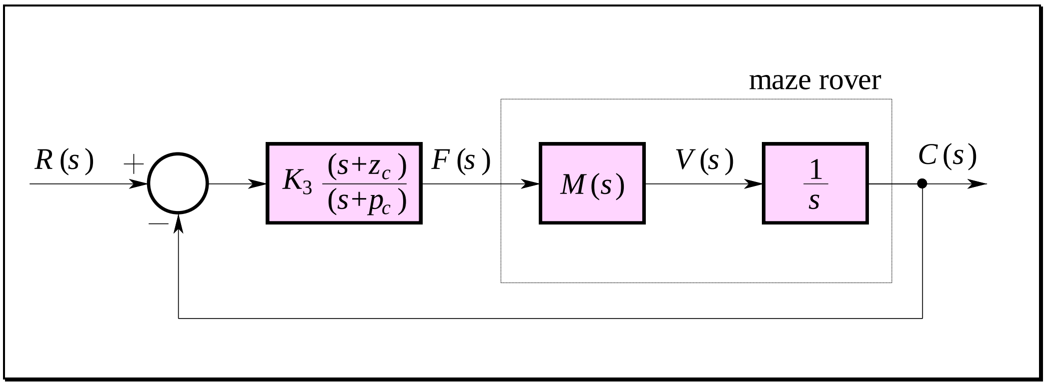



**Figure 3 - Scheme 2 – Phase-Lead Compensator using Root Locus**

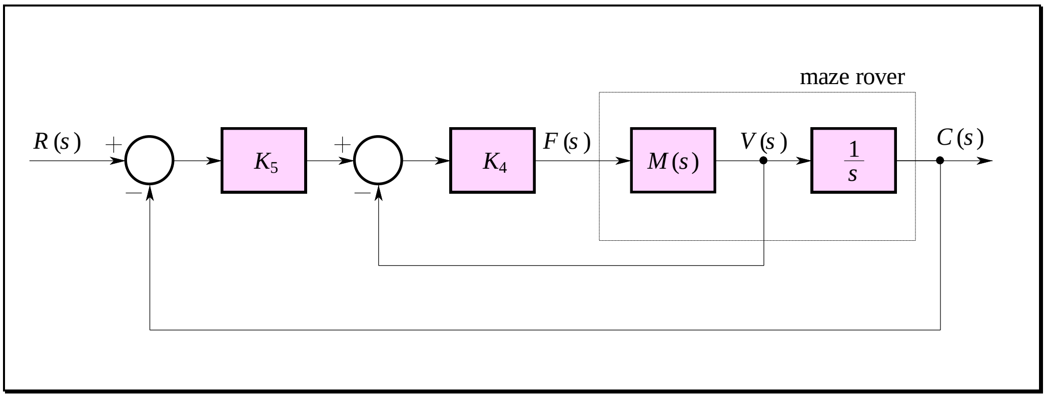



**Figure 4 - Scheme 3 – Minor-Loop Feedback**

Each control scheme will need to be put into the form shown in Figure 1 so that coefficients can be entered into the MR.

# Control.4

## Modelling

All control schemes should be modelled in MATLAB® in *discrete-time*. Use the bilinear transformation to *individually* discretize the continuous time blocks shown in the figures above. Perform the discretization by hand or use the MATLAB® command *c2d*, with the `METHOD` parameter set to `'tustin'`. The control system may need to be adjusted to counter the effect of discretization and quantization (see next section).

## Controller Implementation

In the lab, you will need to download your controller's *discrete-time* coefficients into the MR at the start of every lab session (the controller's coefficients will be erased after every lab session).

The format for the discrete-time versions of $G(s)$ and $H(s)$ are:

$$G_d(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}} \qquad H_d(z) = \frac{d_0 + d_1 z^{-1}}{1 + c_1 z^{-1}}$$

### Setting the controller coefficients

To set the coefficients, follow the steps given below:

It is important to use the "parallel" mode of the dual power supply because the Maze Rover motors draw large transient currents

1. ***Power up the MR with the dual power supply in "parallel" mode.***
2. Connect the MR to the PC via the USB cable.
3. Run the `Maze Rover Interface` Windows program. It is available from:

   http://services.eng.uts.edu.au/pmcl/ss/Downloads/MRInterface.exe

4. To change the controller coefficients, click on the controller buttons. A dialog box will prompt for the controller coefficients.
5. All coefficients need to be *normalised* to 256. That is, before entering coefficients, multiply them by 256 and round the result. This is because the MR has a fixed point processor, so its internal maths uses implied binary points.

## Report

Only submit ***ONE*** report per project group.

A ***HAND WRITTEN*** report with the following sections must be completed:

**Controller Design**

1. Calculation of the **desired pole locations**.
2. **State-variables**. You should perform the design using matrix algebra, and then confirm your design using block diagram reduction. You will have to rearrange your state-feedback to suit the implementation requirements.
3. **Root-Locus**. You should design the controller gain and pole/zero locations by trial and error in MATLAB® using RLTOOL. Allow at least a 10% margin (with respect to the assumed pole value) in any pole-zero cancellation scheme used.
4. **Minor-Loop**. You should design the controller gains by performing block diagram reduction. You will have to rearrange your minor-loop scheme to suit the implementation requirements.

**MATLAB® Simulation**

5. A **continuous-time simulation** of the step-responses of the proposed schemes using MATLAB®. Include an analysis of the actual control specifications achieved (peak time, P.O., settling time).
6. A **discrete-time simulation** of the step-responses of the proposed schemes using MATLAB®. Include an analysis of the effects caused by the bilinear transform and coefficient round-off, and what you did to compensate for them.

**Laboratory Verification**

7. A **plot of experimental results**, for each scheme, verifying that they meet the control specifications.

**The Digital Controller is part of the Group Project.**

# Lab 6 – Control System Modelling

*Maze rover. Step response. Frequency response.*

## Introduction

Before we can control a system using analytical techniques (rather than heuristic techniques like fuzzy control), we need to characterize it. We have already seen the power of the transfer function – it gives insight into both time-domain and frequency-domain behaviour.

Once we know the system transfer function, we can mathematically determine a control strategy to meet performance criteria such as settling time, percent overshoot, etc. This lab explores two complementary methods for synthesising a transfer function from "black box" measurements of the system.

## Objectives

1. To obtain an *s*-domain block diagram model of the maze rover – that is, to synthesize its transfer function by observing its input/output behaviour.

2. To become familiar with the limitations of modelling – measurement errors, assumed linearities, neglecting high-order effects, etc.

## Equipment

- 1 Digital Storage Oscilloscope (DSO) – Agilent DSO-X 2004A with Wave Gen
- 1 Arbitrary Waveform Generator (AWG) – Agilent 33210A with Option 002
- 1 Maze Rover
- 1 DC dual power supply with tracking – GW GPQ-3020

# L6.2

## Maze Rover

The maze rover (MR) was designed to act like a typical vehicle – we apply a force, and a velocity results. Treated as a system that converts force to velocity, we have:

The maze rover's velocity model

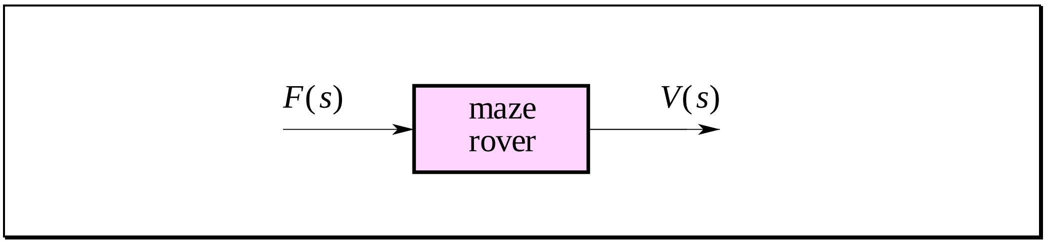



**Figure L6.1**

The force applied to the MR comes from two DC motors controlled by a microcontroller (μC). The input to the system is then really nothing but numbers to command the μC – from a communication channel; from internal software; from a converted analog voltage; or anywhere.

For this lab, there is a "test header" (the 4mm sockets on the bottom PCB of the MR) that allows you to input a voltage in the range -2.5 V to 2.5 V that is treated as the input to the above system. Therefore, the input to the system is not really a force, but a *voltage* to a 10-bit analog-to-digital converter (ADC). The input voltage is interpreted by the μC which then drives the motors.

Similarly, the velocity output is measured by optical rotary encoders that are fed into the μC. The μC outputs the velocity measurement as a 10-bit pulse width modulated (PWM) *voltage* waveform, which is lowpass filtered by a simple first-order active filter. This is a simple implementation of a digital to analog converter (DAC). One socket on the test header outputs the analog voltage representing the velocity.

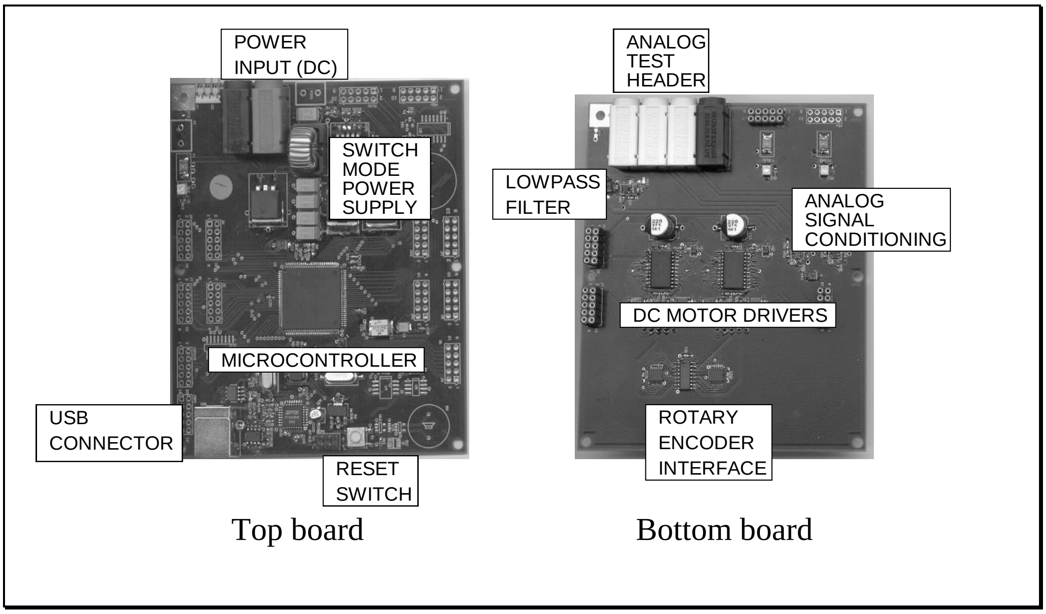
Therefore, even though the MR model has "force in / velocity out", for modelling purposes we are able to treat the MR as a "voltage in / voltage out" device.

## Setting Up

We need an easy way to characterize the MR without getting involved in the software and the dynamics of the DC motors. The MR test header allows direct input / output style measurements of the MR transfer function.

### In the Lab – Setting up

1. Examine the printed circuit boards (PCBs) that make up the MR. The top PCB is the μC board. The bottom PCB is the motor driver board. The MR is powered via a switch mode power supply (SMPS) on the μC board. It can be powered by either a laboratory power supply (for testing purposes) or by batteries (for remote operation). Identify the following components:




The maze rover's microcontroller and motor driver printed circuit boards

**Figure L6.2**

2. Make sure the name of your MR starts with the same letter as your specification. You may need to move benches to find your MR.

3. **Make sure the MR is sitting up vertically so that its wheels are free to spin.** You will be conducting a velocity test, and you don't want the MR to travel along the laboratory bench!

# L6.4

The maze rover's power and interface connections

4. Use the laboratory power supply to provide power to the MR. It requires a DC voltage in the range 9-13V. **Use a 12V DC supply**:

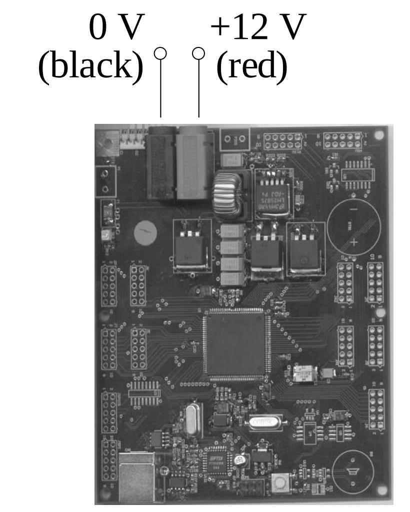



**Figure L6.3**

5. The analog test header provides for two inputs and one output. *One input is not used in this lab.* **The black 4mm connector is to be connected to the FG and DSO ground.** The yellow 4mm connector next to the black 4mm connector is the function generator (FG) input. The outermost yellow 4mm connector is the output (observe on the DSO):

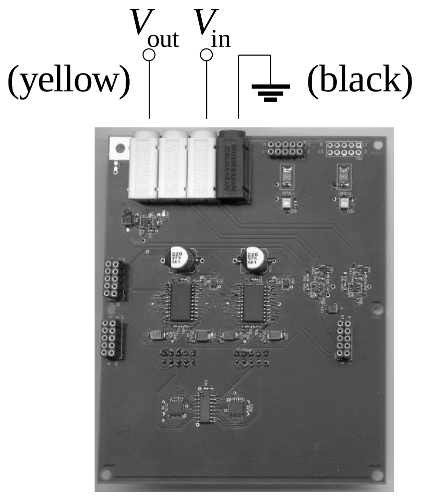



**Figure L6.4**

6. Note that $V_{in}$ from the analog test header gets converted to a 10-bit number via a ratiometric ADC in the microcontroller. Since the μC uses a unipolar ADC, it expects signals in the range 0–5 V. The analog interface circuitry on the motor driver board offsets the analog input by 2.5 V. Therefore:

**<u>The analog input range is –2.5 V to +2.5 V</u>.**

7. Note that $V_{out}$ from the analog test header is in the range 0–5 V. You will need to *mentally* subtract 2.5 V from all your output measurements (by, for example, simply moving the trace on the DSO down 2.5V). Therefore:

**<u>The analog output range is 0 V to +5 V</u>.**

## System Characterization

Treating the MR as a "black box", we can obtain the step response and frequency response. We can then characterize the MR by a transfer function. The transfer function will only be approximate, because it is derived under certain conditions – for example, it is assumed that the MR is linear and time-invariant and remains within this linear region of operation; there are measurement errors and noise on the input and output signals; etc.

### In the Lab – Step Response

1. Set up a *very* low frequency **4 V peak-to-peak** (-2V to +2V) square wave (around 0.1 Hz) on the FG.

We're just making a "step" using a square wave

2. Measure the *step response* of the MR.

### In the Lab – Frequency Response

1. After analysing the step response, you should be able to ascertain a range of frequencies that would be useful for a frequency response (Bode plot). The frequencies will be very low, since we're dealing with a mechanical system with a large inertia.

Remember – a frequency response is obtained with a sinusoid

2. Use a **4 V peak-to-peak** (-2V to +2V) sinusoid on the FG.

3. Measure the *frequency response* of the MR.

# L6.6

## Maze Rover Transfer Function

The test header takes an analog voltage and provides it to the ADC of the μC. The output velocity is measured and sent to the DAC of the μC.

Since we want to model the MR just as a force input and a velocity output, we need to take into account the analog test circuitry and the units of the μC's output. The actual test setup has the following block diagram:

The maze rover's test model

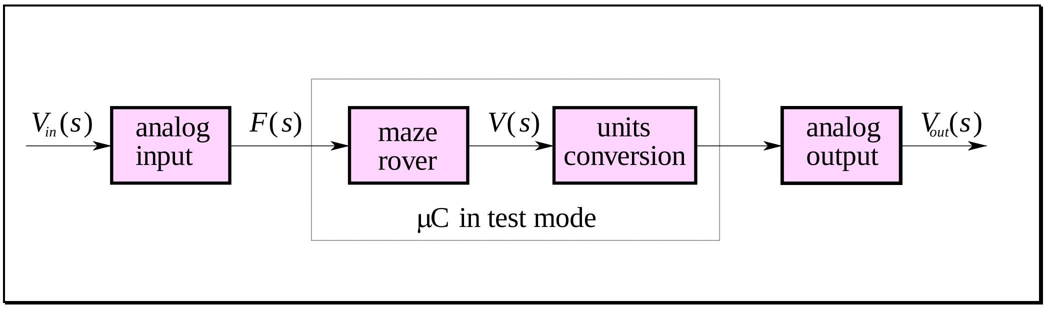



**Figure L6.5**

The analog input and output transfer functions can be considered the inverses of one another, apart from a DC shift in the analog output, i.e. they can effectively be ignored.

Due to the way the software in the μC has been designed, there is need for a conversion of the velocity units from "distance per second" to "distance per sample period". For example, with a sample period of $T_s = 1\,\text{ms}$, the μC will supply a velocity value that is "distance per ms". In this case the MR transfer function in test mode would be modelled as:

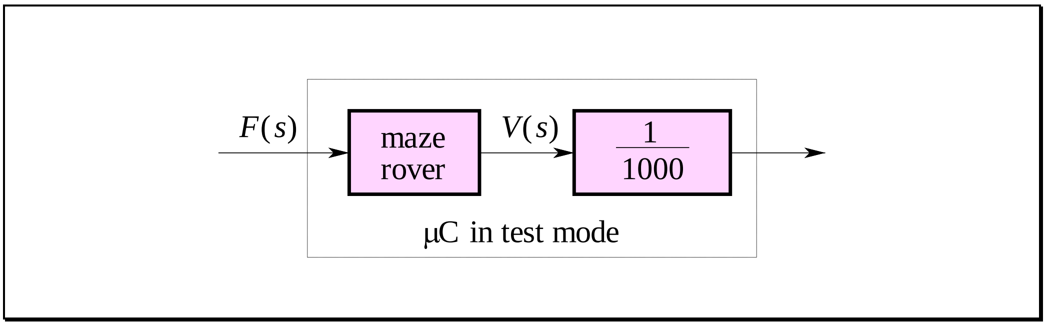



**Figure L6.6**

For this example, the maze rover transfer function is derived from the experimental data by multiplying all output values by 1000.

## Report

Only submit ***ONE*** report per project group.

A ***HAND WRITTEN*** report with the following sections must be completed:

**Experimental Results**

1. Outline the procedure used in modelling the MR transfer function from the step response. Graph the experimental results.
2. Outline the procedure used in modelling the MR transfer function from the frequency response. Graph the experimental results.

**Transfer Function**

3. Model the MR with a transfer function, treating the input as force, and the output as velocity. Determine the transfer function using the results from the step response and/or frequency response.

**MATLAB® Verification**

4. Provide a MATLAB® plot of a theoretical **step response** to verify the proposed MR model, showing relevant parameters.

   Provide a MATLAB® plot of a theoretical **frequency response** to verify the proposed MR model, showing relevant parameters, **together with experimental results on the same graph**.

**Lab 6 is part of the Group Project.**

# Group Project – Demodulator Design

*Phase-locked loop. Coherent demodulation. Filter design.*

## Introduction

The MR uses a wireless communication system to allow movement commands to be issued remotely. This document outlines the communication system specifications. The control system is specified in a separate document.

## Communication System Overview

The communication scheme is used to achieve remote operation:

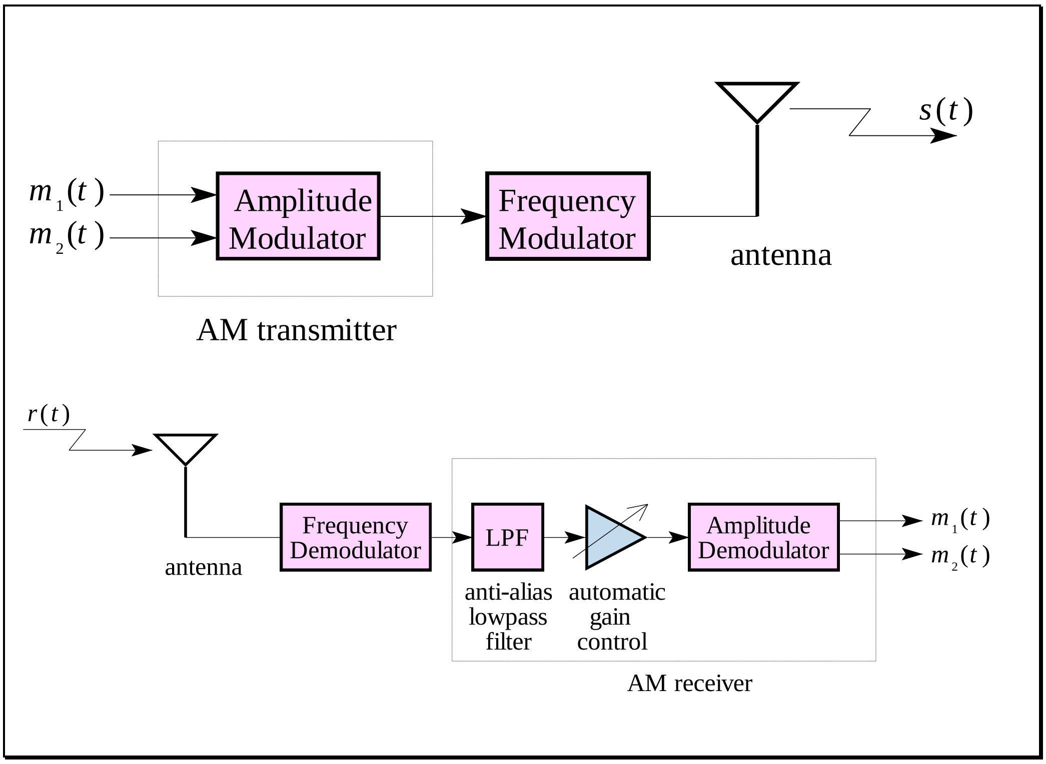



**Figure 1 - General Communication Scheme**

The presence (or absence) of 2 set frequencies (sinusoids) is how the MR receives digital commands. There are thus 4 commands possible: forward, left, right and "do nothing".

The PC is the transmitter. You have to design the MR receiver according to the following nominal spectrum allocations and message signals.

# Comms.2

## Maze Rover Communication System

The MR has motors that drive it forward and turn it left or right. The motors need to be controlled so that the MR moves in "steps" as it navigates a maze. There is an on-board communication system designed to receive commands from a user – either forward, left, right or do nothing. The communication scheme has to be designed to receive specific signals that are transmitted from a base station by the user.

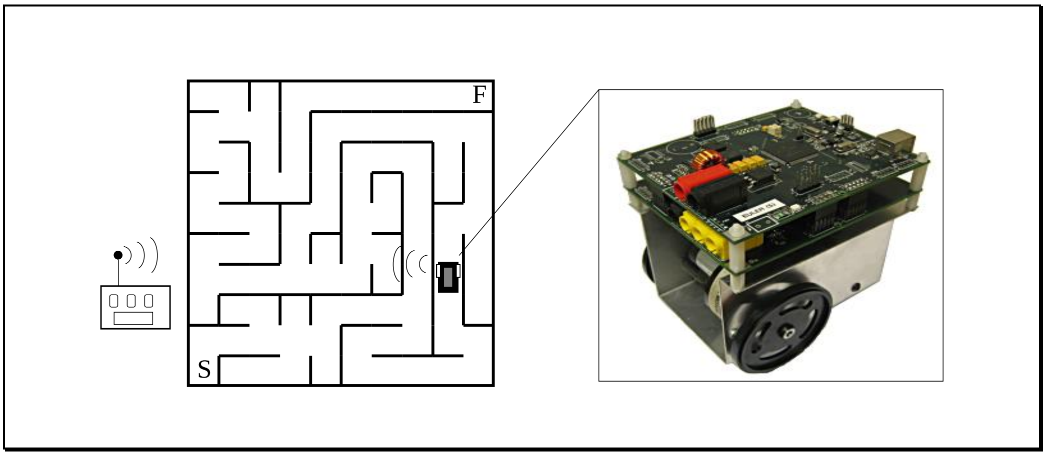



**Figure 2 – Maze Rover in maze**

Since there are only 4 messages (or *symbols*) to send, these can be encoded using 2 bits. The mapping between commands and bits is shown below:

| **Bits** | **MR Command** |
|---|---|
| 00 | Stop (do nothing) |
| 01 | Turn left |
| 10 | Turn right |
| 11 | Move forward |

The MR communication system uses a simple digital transmission technique known as *amplitude-shift keying* (ASK) or *on-off keying* (OOK). This simple technique assigns the value of a bit (0 or 1) to the presence or absence of a sinusoid.

An example of a single binary message that is amplitude-shift keyed is shown below:

An ASK signal

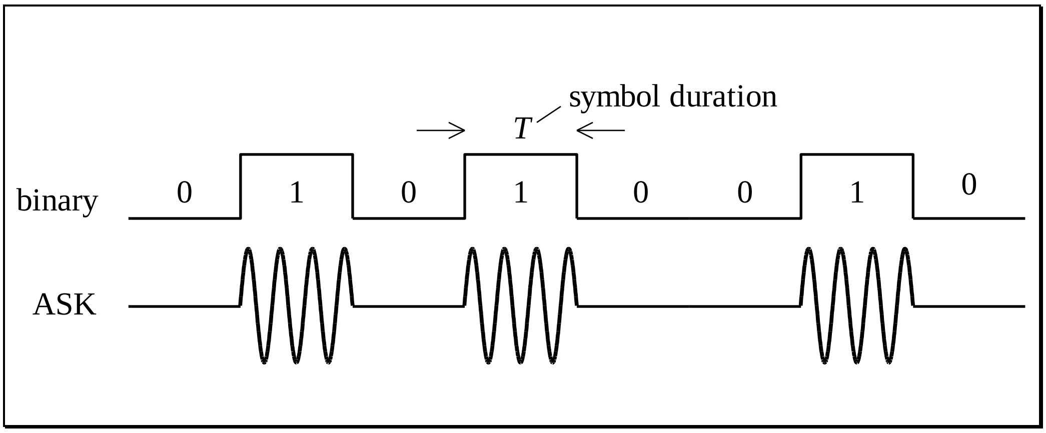



**Figure 3 – Amplitude-shift keying signal**

The binary signal simply turns on or off a carrier (sinusoid). To transmit two bits, we need 2 "ASK carriers", designated as $m_1(t)$ and $m_2(t)$.

Since there are numerous MRs, each MR has been allocated a *channel* in the available electromagnetic spectrum. Channel allocation ensures that MRs can communicate simultaneously.

Channel allocation

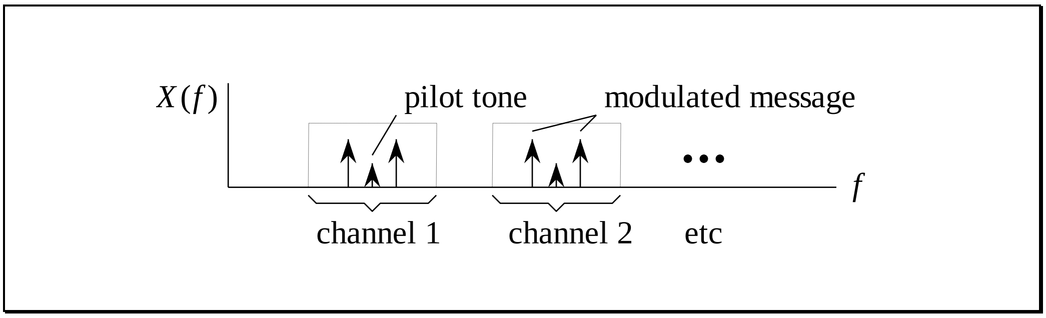



**Figure 4 – Communication channels**

To shift the message signals to their appropriate channel, the messages undergo amplitude modulation:

Amplitude modulation of the message signals

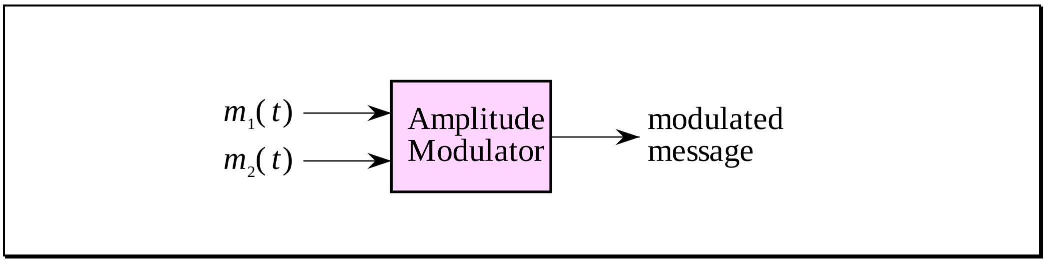



**Figure 5 – Amplitude modulation of the messages**

# Comms.4

## Frequency Specifications

All MRs have a sample frequency of $f_s = 12500$ Sa/s.

Individual MRs must meet the following specifications in terms of message and carrier frequencies.

| Group Project | Message 1 Frequency, $f_1$ (Hz) | Message 2 Frequency, $f_2$ (Hz) | Carrier Frequency, $f_c$ (Hz) |
|---|---|---|---|
| A | 450 | 400 | Experimentally determined to nearest harmonic of 25 Hz. |
| B | 350 | 300 | |
| C | 250 | 200 | |
| D | 150 | 100 | |
| E | 100 | 150 | |
| F | 200 | 250 | |
| G | 300 | 350 | |
| H | 400 | 450 | |
| I | 400 | 350 | |
| J | 300 | 250 | |
| K | 200 | 150 | |
| L | 100 | 50 | |

### Modulated Waveform Measurements (Lab 7)

You are to experimentally determine the type of modulation used.

You are to experimentally determine the nominal carrier frequency.

## Demodulation Scheme

The following demodulation scheme is to be designed and experimentally tested.

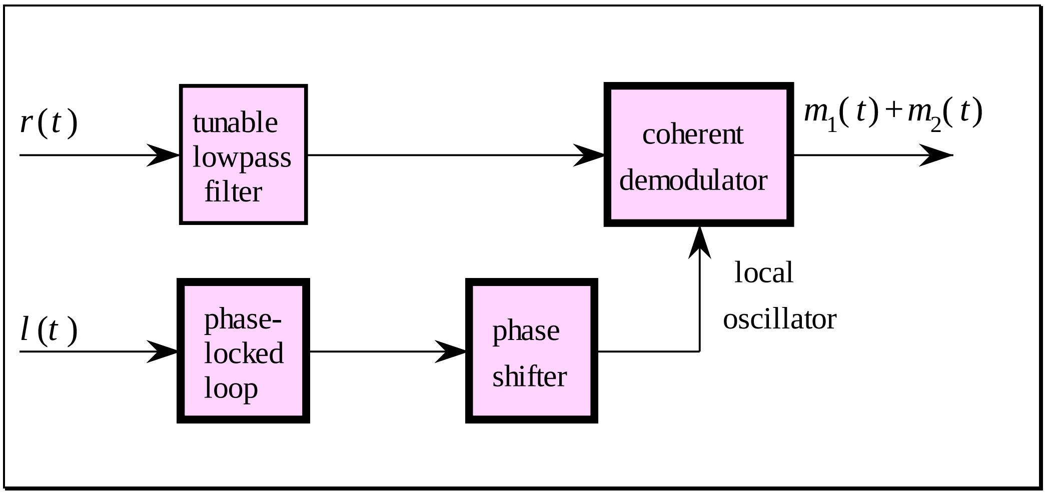



**Figure 6 - Demodulation Scheme**

The received message signal, $r(t)$, is passed through an *analog* tuneable lowpass filter which implements the anti-alias filter and automatic gain control (AGC) functions.

The signal $l(t)$ is a pilot tone (small carrier) that has been extracted from the modulated signal. The pilot tone is necessary to synchronise the receiver's local oscillator to the transmitter's carrier (if there is a frequency or phase error at the receiver, the recovered signals will be distorted and attenuated). The pilot tone is used by the phase-locked loop (PLL).

The phase-locked loop is a feedback control system used to "lock onto" a pilot tone in the modulated waveform, and thus generate a local oscillator that is in perfect frequency and phase synchronism with the transmitter.

The phase shifter is used to introduce a phase shift in the output of the PLL, so that it has the correct phase with respect to the pilot tone in the modulated waveform. **The output of the phase shifter is the local oscillator.**

The coherent demodulator uses the local oscillator to demodulate the transmitted signal, producing the original message signals.

# Comms.6

## Phase-Locked Loop Design

The phase-locked loop should be designed using the theory in Appendix B as a *guide*. The block diagram below shows the design of the PLL:

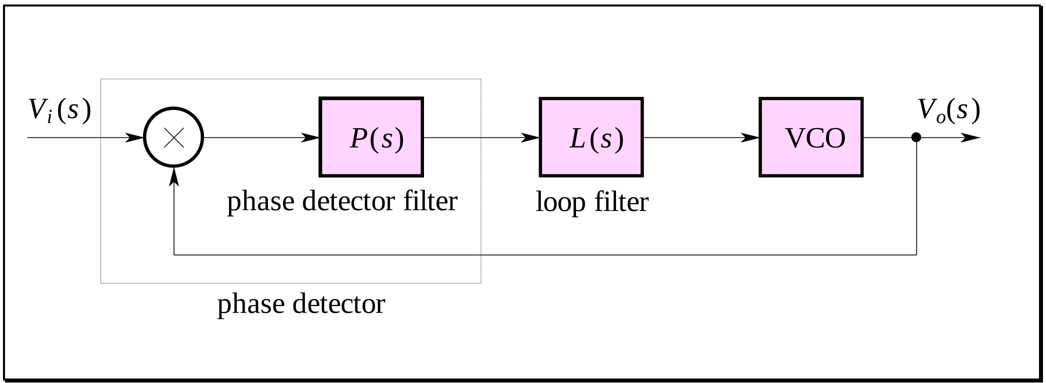



**Figure 7 – Phase-Locked Loop Scheme**

### Phase Detector

The phase detector is normally just a multiplier – but this assumes that the high frequency term is removed by the loop filter. For PLLs operating at a fairly low frequency (kHz), this assumption is not justified. Therefore, the phase detector needs to be implemented as:

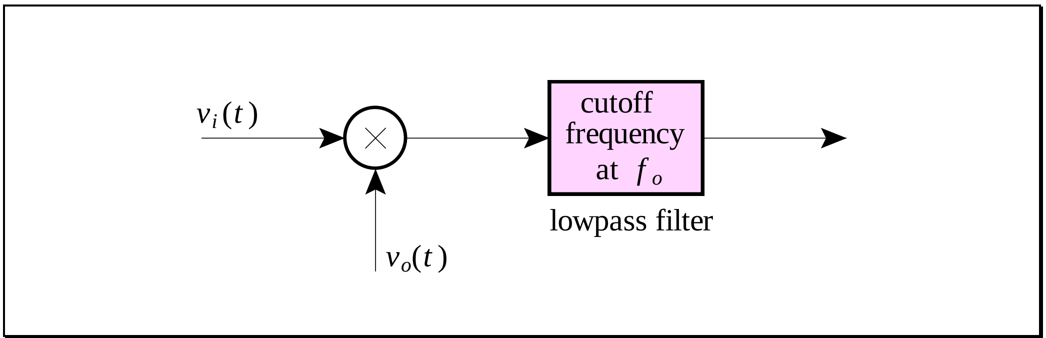



**Figure 8 – Phase Detector**

Design the phase detector's lowpass filter to be first-order with a cutoff frequency equal to the nominal VCO frequency. This will ensure adequate attenuation of the high frequency term coming out of the multiplier.

You should use the method of "impulse-response matching" to design the filter. The gain of the resulting discrete-time filter should be adjusted so that the DC gain is 1.

The gain of the multiplier is such that $K_0 = 2/\pi$ .

### Loop Filter

The phase-locked loop is to be implemented with a loop filter of the form:

$$L(s) = \pm \frac{1 + s\tau_2}{1 + s\tau_1}$$

Using the bilinear transform, it can be shown that the equivalent discrete transfer function in the *z*-domain has one pole and one zero.

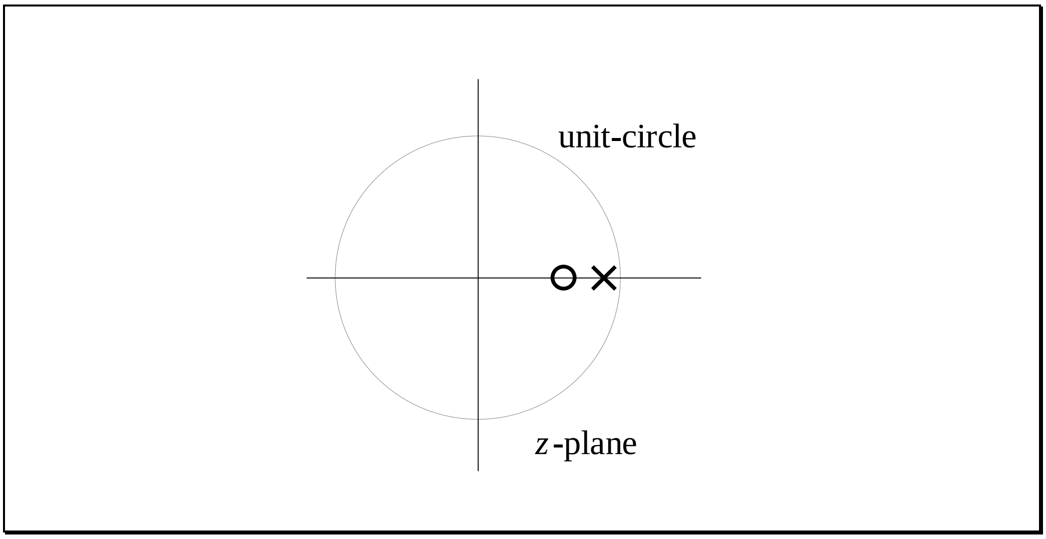



**Figure 9 – *z*-plane Pole-Zero Plot**

As a safety margin for preventing round-off error pushing the pole outside the unit-circle, it is desired to have the *z*-plane pole at 0.9, and then to derive $\tau_1$ from this.

To obtain a suitable value for $\tau_2$ that gives optimum transient performance, set the overall closed-loop damping ratio to $\zeta = 1$.

Design the loop filer using the bilinear transform.

### VCO Characteristic (Lab 7)

You are to measure the VCO's characteristic ( $f_o$ and $k_v$ ).

# Comms.8

## Phase Shifter Design

The loop filter used in this PLL is different to the loop filter given in Appendix B. Analyse the response of your PLL to a constant frequency and phase error in the incoming pilot signal, and *design* an appropriate phase shift to apply to the output of the PLL so that the resulting local oscillator is in perfect frequency and phase synchronism with the pilot tone. The phase shift should be calculated to within one degree.

## Demodulator Design

The phase-locked loop provides the receiver with a local oscillator that is in synchronism with the transmitter, allowing coherent demodulation of the signal.

You are to design and test a coherent demodulator. Any filters required in the demodulator are to be implemented by cascading a resonator and a comb filter.

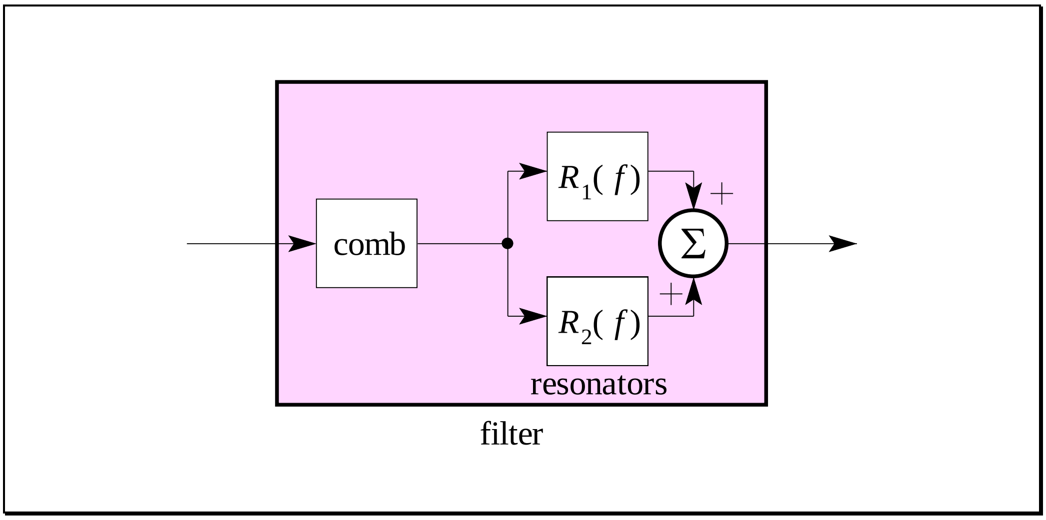



**Figure 10 – Comb-resonator filter**

Details for the comb and the resonator are outlined in the following sections.

You need to design all coefficients (including the attenuation) for each resonator.

## Comb Filter

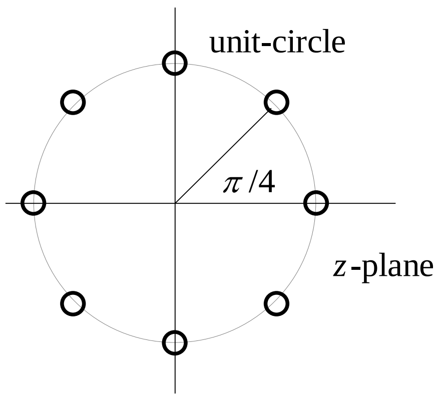
The comb filter is an example of a "Finite Impulse Response", or FIR, filter. FIR filters do not have any poles (hence their finite impulse response). One type of comb filter is described by the transfer function:

$$H(z) = 1 - z^{-m}$$

We find the zeros of the transfer function by setting $H(z) = 0$. This gives:

$$\begin{aligned} z^{-m} &= 1 \\ z^{m} &= 1 \\ \left(e^{j\theta}\right)^{m} &= e^{j2n\pi} \\ \theta &= \frac{2n\pi}{m}, \quad n = 1,2,\ldots m \end{aligned}$$

For example, if $m = 8$, then $\theta = \frac{n\pi}{4}$. A plot of the zeros in the *z*-plane is then:



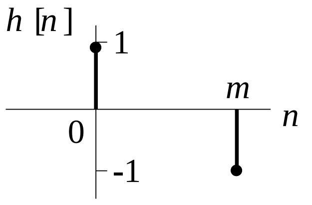
# Comms.10

## Unit-Pulse Response

Taking the inverse $z$-transform of the transfer function, we get the unit-pulse response:

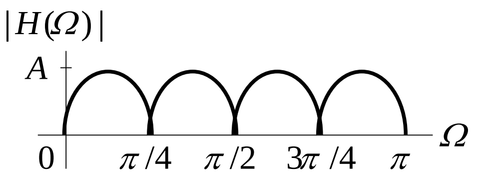
$$h[n] = \delta[n] - \delta[n-m]$$

which looks like:




## Frequency Response

The frequency response is obtained by setting $z = e^{sT_s} = e^{j\omega T_s} = e^{j\Omega}$ in the transfer function. A graph of the magnitude response for $m = 8$ looks like:




## Difference Equation

Finally, the difference equation for the comb is given by:

$$y[n] = x[n] - x[n-m]$$

## MR Implementation

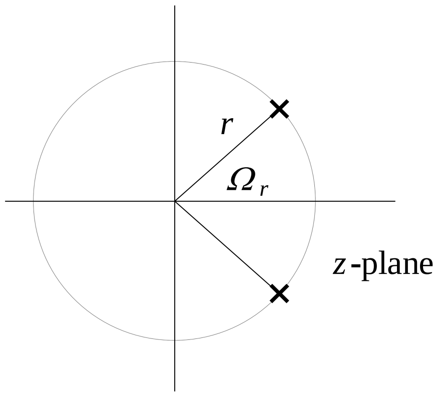
The MRs have a comb filter with $m = 250$.

## Resonator

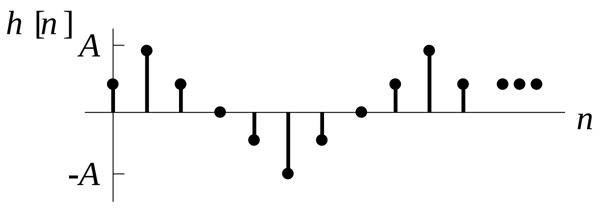
A discrete-time resonator produces a sinusoid when "hit" with a unit-pulse. It's transfer function is:

$$H_R(z) = \frac{z^2}{z^2 - 2r\cos\Omega_r z + r^2} = \frac{z^2}{\left(z - re^{j\Omega_r}\right)\left(z - re^{-j\Omega_r}\right)}$$

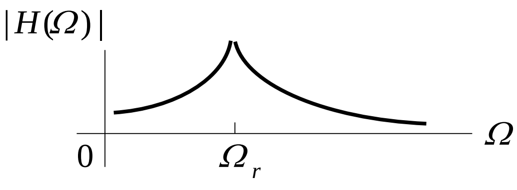
A plot of it's poles is:




### Unit-Pulse Response




### Frequency Response




### Difference Equation

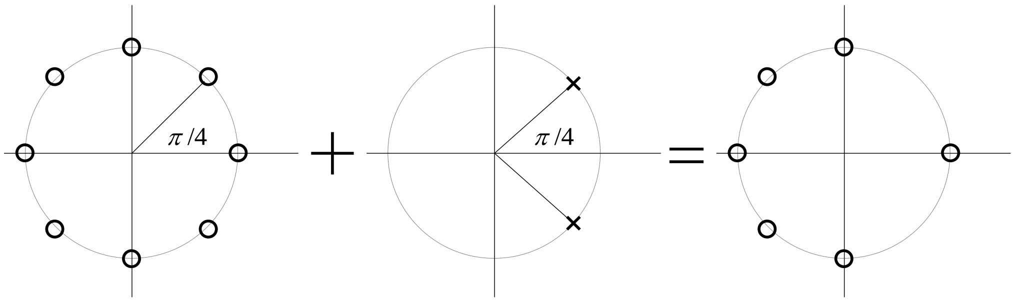
$$y[n] = 2r\cos\Omega_r y[n-1] - r^2 y[n-2] + x[n]$$

We normally take $r = 0.99999 \approx 1$ to keep the system stable.

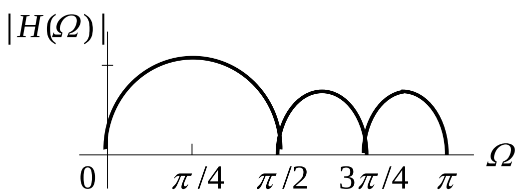
# Comms.12

## Comb-Resonator Combination

The idea behind cascading the comb and resonator is for the resonator's pole pair to approximately cancel out one pair of the comb's zeros. For example:




The resulting frequency response, in this case, makes a bandpass filter:




### Resonator Attenuation

The resonators produce very large outputs and need to be attenuated – they are implemented with an attenuation factor, $r_0$, as shown by the transfer function:

$$R(z) = \frac{1}{r_0}\left(\frac{1}{1 + r_1 z^{-1} + r_2 z^{-2}}\right)$$

To design an appropriate value for the attenuator, you will need to simulate the comb-resonator combination in MATLAB® and observe the output after 40ms of excitation (40 ms is the period of a communication *symbol*). For example, an output peak of 350 after 40 ms of excitation, for an input peak of 1, would require an attenuation factor of $r_0 = 350$.

## Demodulator Implementation

The MR needs to be controlled from a PC. Follow the steps below to control the functionality of the MR.

### Setting up the MR

1. Connect the MR to the PC via the USB cable.
2. Run the `Maze Rover Interface` Windows program. It is available from:

   http://services.eng.uts.edu.au/pmcl/ss/Downloads/MRInterface.exe

### Setting filter coefficients

1. To change the filter coefficients, click on the relevant filter button. A dialog box will prompt for the filter coefficients.
2. All coefficients need to be *normalised*. The normalising factor is given next to the equations below. For example, for the phase detector filter, multiply the coefficients by 256 and round the result.

The MR has a fixed point processor, so its internal maths uses implied binary points

### Phase Detector Filter

The format for the discrete-time version of the phase detector filter is:

$$P_d(z) = \frac{f_0 + f_1 z^{-1}}{1 + e_1 z^{-1}}$$

Normalize to 256

### Loop Filter

The format for the discrete-time version of the loop filter is:

$$L_d(z) = \frac{h_0 + h_1 z^{-1}}{1 + g_1 z^{-1}}$$

Normalize to 256

### Phase Shifter

The phase shifter needs an integer value between -180 and 180 degrees.

Integer between -180 and 180 only

### Resonator

The format for the discrete-time version of the resonator is:

$$R(z) = \frac{1}{r_0}\left(\frac{1}{1 + r_1 z^{-1} + r_2 z^{-2}}\right)$$

Normalize $r_1$ and $r_2$ to 16384, normalize $r_0$ to 1

# Comms.14

## Report

Only submit ***ONE*** report per project group.

A ***HAND WRITTEN*** report with the following sections must be completed:

**PLL Design**

1. Outline the procedure used in determining the VCO's characteristic. Graph the experimental results. Determine the VCO characteristic.

2. Detail the theoretical design of the phase detector filter.

3. Detail the theoretical design of the loop filter. Detail the design of the phase shifter.

**Demodulator Design**

4. Outline the procedure used in determining the modulation technique. Include relevant sketches of experimentally obtained spectra and time-domain waveforms. Determine the modulation technique. Determine the MR's carrier frequency.

5. Design (by hand) your demodulator, showing magnitude spectra at each point in the system.

6. Detail the theoretical design of the demodulator's comb-resonator filter. Provide a MATLAB® simulation of the entire demodulator.

**Laboratory Verification**

7. Implement the PLL and demodulator in the laboratory, and show that it demodulates any combination of message signals.

**The Demodulator Design is part of the Group Project.**

# Lab 7 – Communications Testing

*Modulated signals. VCO characteristic.*

## Introduction

There are many types of amplitude modulation used in practice – such as AM, DSB-SC, SSB and QAM. The design of a demodulator depends on the modulation technique used. Observation of a modulated waveform in both the time- and frequency-domains while sending various "baseband" message signals allows a modulation technique to be determined.

A coherent demodulator relies on producing a local oscillator that is in phase and frequency synchronism with the transmitter. A phase-locked loop (PLL) is often used in a modern receiver to regenerate a local carrier sinusoid for use in the demodulation process. One part of the PLL is a voltage controlled oscillator (VCO). To design the PLL to meet certain time- and frequency-domain specifications, we need to ascertain the relationship between the VCOs input voltage and the frequency of its output sinusoid.

## Objectives

1. To determine the modulation technique used by the maze rover's communication system, and the nominal carrier frequency.

2. To characterize the digital VCO used in the PLL of the maze rover's communication system.

## Equipment

- 1 Digital Storage Oscilloscope (DSO) – Agilent DSO-X 2004A with Wave Gen
- 1 Arbitrary Waveform Generator (AWG) – Agilent 33210A with Option 002
- 1 Maze Rover
- 1 TIMS trainer
- 1 PC running Windows XP (or later version) with Internet connection
- 1 DC dual power supply with tracking – GW GPQ-3020

# L7.2

## Amplitude Modulator

We need an easy way to observe the modulated waveforms that the MR is receiving. The Maze Rover Interface program running on a Windows PC allows test messages to be sent to the MR. The MR test header outputs the received modulated signal.

### In the Lab – Setting up

1. Make sure the name of your MR starts with the same letter as your specification. You may need to move benches to find your MR.

2. Use the laboratory power supply to provide power to the MR. It requires a DC voltage in the range 9-13V. **Use a 12V DC supply**:

The maze rover's power and interface connections

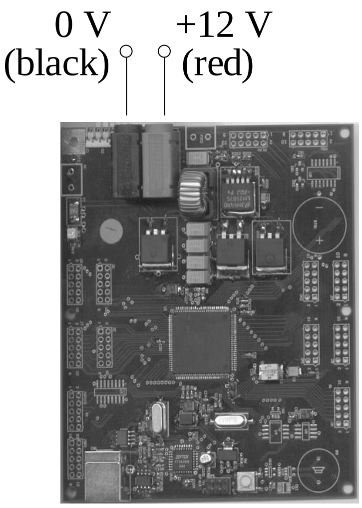



**Figure L7.1**

3. Connect the MR to the PC via the USB cable.

4. Run the `Maze Rover Interface` Windows program. It is available from:

   http://services.eng.uts.edu.au/pmcl/ss/Downloads/MRInterface.exe

5. The analog test header provides for two inputs and one output. *One input is not used in this lab.* **The black 4mm connector is to be connected to the DSO ground.** The outermost yellow 4mm connector is the output (observe on the DSO):

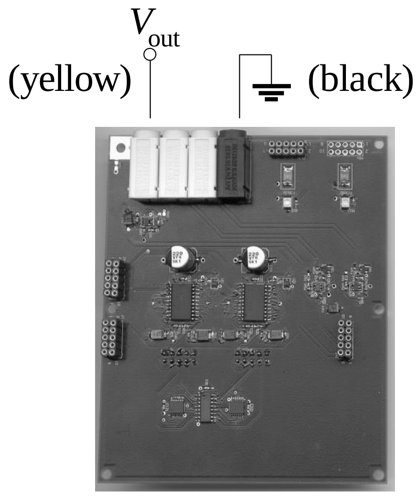



**Figure L6.1**

6. Note that $V_{out}$ from the analog test header is in the range 0–5 V. You will need to *mentally* subtract 2.5 V from all your output measurements (by, for example, simply moving the trace on the DSO down 2.5V). Therefore:

**The analog output range is 0 V to +5 V.**

**In the Lab – Modulation Technique**

1. From the Maze Rover Interface program running on the PC, select `Test Message` from the `Comms Mode` drop-down list box. You may then select a message from the `Message` drop-down list box.

2. Observe the modulated waveforms on the DSO in both the time- and frequency-domains to determine the type of amplitude modulation used to shift your message signals to the correct channel.

3. Determine the MR's carrier frequency to the nearest 25 Hz or 50 Hz (the figure given in the specification).

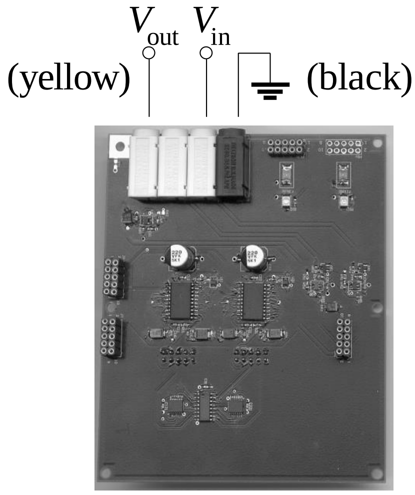
# L7.4

## VCO Characterization

To measure the VCOs input-output characteristic in Hz/V, a DC voltage is applied to the MR's test input and the frequency (not the amplitude) of the resulting output sinusoid is measured on the MR's output.

### In the Lab – VCO Characteristic

1. The analog test header provides for two inputs and one output. *One input is not used in this lab.* **The black 4mm connector is to be connected to the TIMS and DSO ground.** The yellow 4mm connector next to the black 4mm connector is the DC input. Use the variable DC supply located at the bottom left-hand side of the TIMS front panel. The outermost yellow 4mm connector is the output (observe on the DSO):




2. On the Maze Rover Interface program, changing the MR `Comms Mode` to `Test VCO` allows you to apply a DC voltage to the MR input and observe the VCO output from the MR output.

3. Measure the VCO's characteristic.

**<u>Lab 7 is part of the Group Project.</u>**