TELE4653 – Digital Modulation and Coding

Lab 1. Quadrature Amplitude Shift Keying (QASK)

Total marks: 5 marks

1. Lab introduction

Quadrature Amplitude Shift Keying (QASK) is also known as quadrature phase shift keying (QPSK). QASK (QPSK) is the most-widely used digital modulation technique. It is used in wireless communication systems such as WLANs/Wi-Fi and LTE. Compared with BPSK, QASK (QPSK) extends the binary phase to 4-ary phase (i.e., 0, pi/2, pi, 3pi/2 rad). In this laboratory exercise, modulation & demodulation of QASK (QPSK) will be the topic.

This lab consists of two experiments, namely Experiment 1– QASK Transmitter, and Experiment 2 – QASK Receiver. The experiments are carried out in the platform of Matlab-Simulink. You can access Matlab 2019b in UNSW myAccess without installing the software in your own computer. For instructions, please see Appendix A.

Simulation source files, i.e., QASKTransmitter.SLX, QASKReceiver.SLX, and video demonstration are provided to assist you in this lab. What you need to do with the provided source files is to set the parameters as per Quiz Questions and observe the output. A report that addresses Quiz Questions is expected as the output of this lab and based on which a mark will be given as part of the assessment of this course.

2. Experiment instructions

This lab consists of two experiments, namely Experiment 1 – QASK Transmitter, and Experiment 2 – QASK Receiver. Simulation source code and video demonstration are provided to assist you in this lab.

(a) Experiment 1 – QASK Transmitter
Open the file QASKTransmitter.SLX in MATLAB 2019b, which is available in the link below.

https://weiwang-wys.github.io/files/Lab1/QASKTransmitter.slx

This experiment investigates the transmitter/modulator of QASK. The data rate of QASK signal is set as R = 2000 bps, and centre frequency is set as fc = 8kHz. For detailed instruction of this experiment, please watch the following video.

https://weiwang-wys.github.io/files/Lab1/QASKTransmitter.mp4

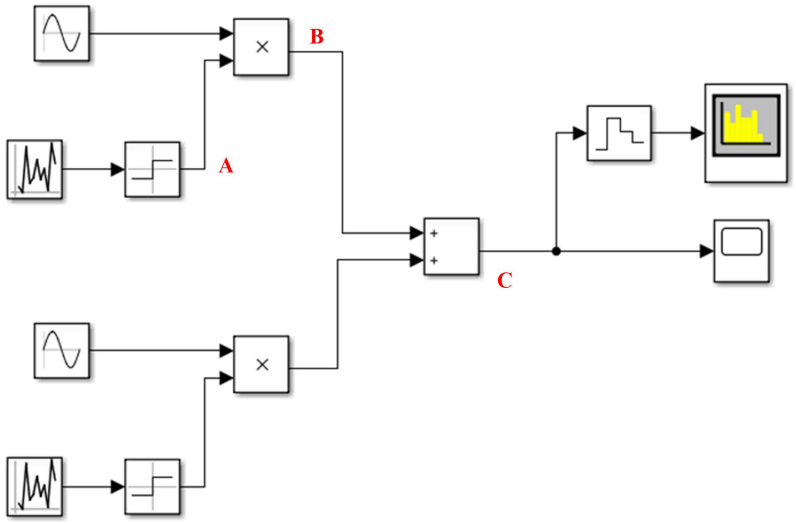


Fig. 1 Block diagram of QASK transmitter

(b) Experiment 2 – QASK Receiver

Open the file QASKReceiver.SLX in MATLAB 2019b, which is available in the link below.

https://weiwang-wys.github.io/files/Lab1/QASKReceiver.slx

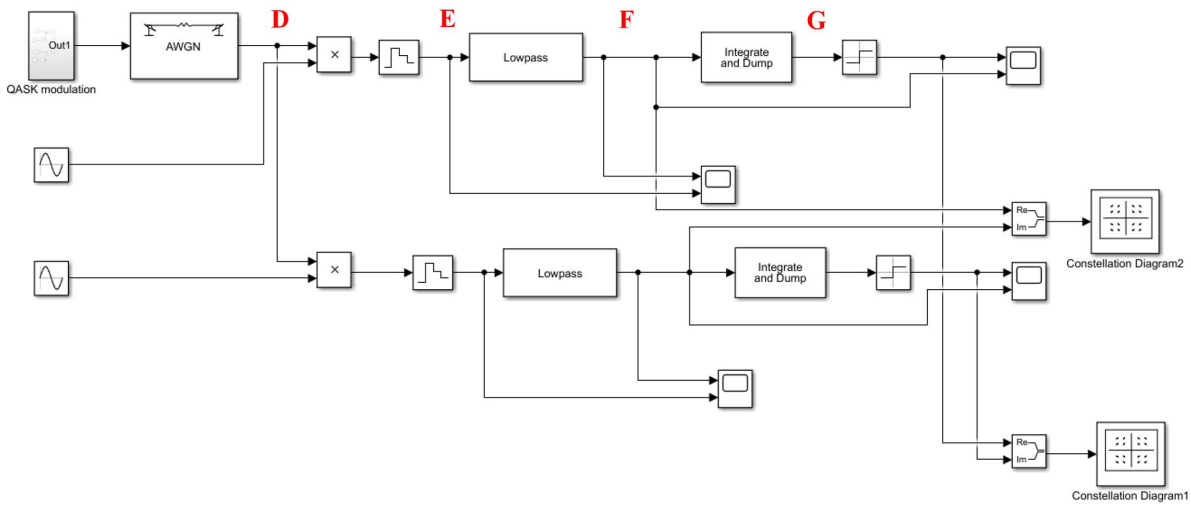


Fig. 2 Block diagram of QASK receiver

This experiment investigates the receiver/demodulator of QASK signal. For detailed instruction of this experiment, please watch the following video.

https://weiwang-wys.github.io/files/Lab1/QASKReceiver.mp4

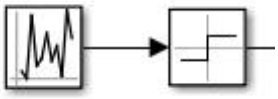
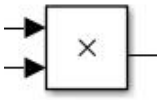
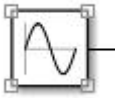
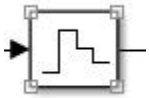
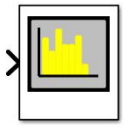
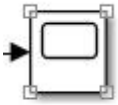
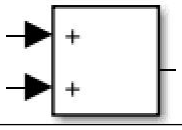
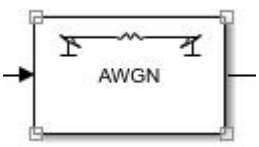
Based on Experiment 1&2, please answer the questions below in your lab report.

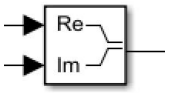
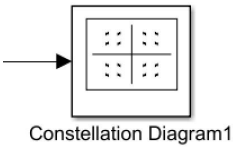
Quiz Question 1. Observe the waveform (time domain) and spectrum (frequency domain) of the points A to G in Fig. 1 and Fig. 2. Please paste the spectrums in your report and discuss their differences. (hint: you can compare bandwidth, centre frequency, and the number of main lobes)

Quiz Question 2. What is the purpose of deploying a lowpass filter after the multiplier in Fig. 2? What is the feasible range of the lowpass filter's cutoff frequency (assuming that the frequency response of the lowpass filter is of the "ideal" rectangular shape)?

Quiz Question 3. Observe the constellations in constellation diagram 1&2 in Fig. 2 when the default phases of the two sine waves are (0, pi/2). Then, set the phases as (0+pi/6, pi/2+pi/6) and (0-pi/6, pi/2–pi/6), and observe the corresponding constellations. Please paste the observed constellations in your report and discuss the impacts of phase shift on the observed constellations.

Appendix A-- Description of modules

	Binary sequence generator which generates random binary sequence with the values being -1 and 1.
	Multiplier
	Sine wave generator
	Zero-order hold, which works as a sampler (a.k.a ADC, analog-to-digital converter)
	Spectrum Analyzer
	Scope
	Adder
	Lowpass Filter
	Integrate and Dump: Integrate over the number of samples in the integration period and reset at the end of the integration. The integrate-and-dump circuit is a *correlator* whose output equals the matched filter output **only at the sampling instants**, and not in-between.
	Additive white Gaussian noise (AWGN) channel

	Real-Image to Complex Converter, which constructs a complex output from real and/or imaginary input.
	Constellation diagram