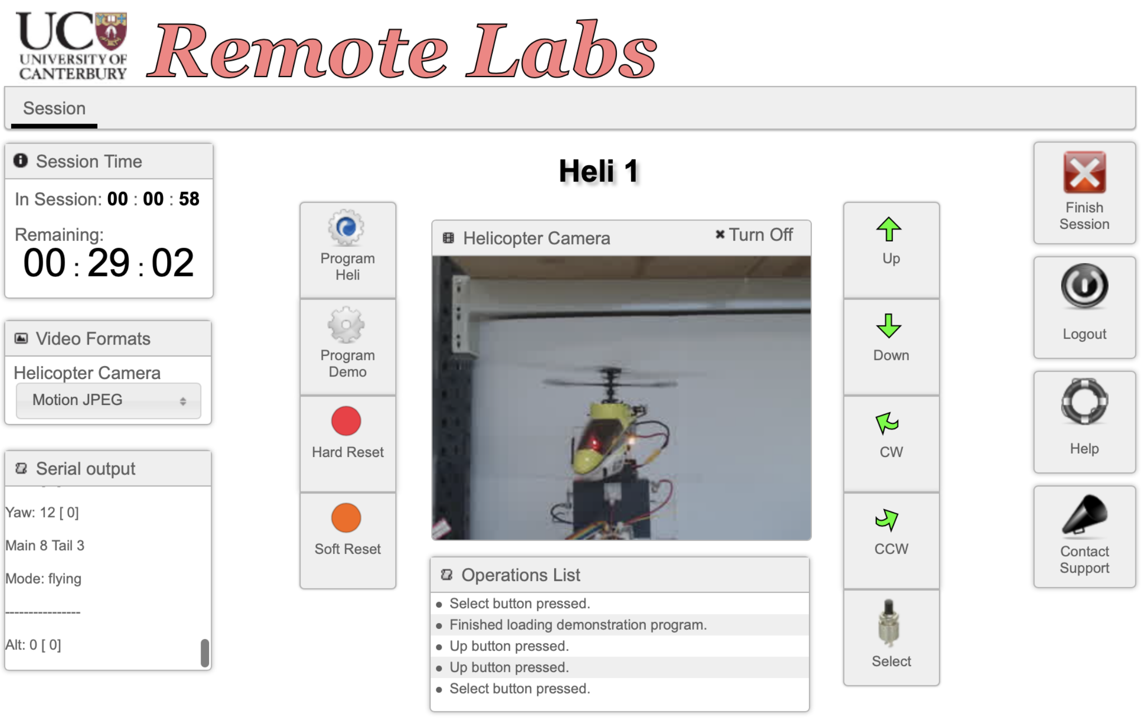
ENCE361: Helicopter Rig Controller Project

Introduction

The goal of this project is to program a remote controlled helicopter so that a "pilot" can fly a sortie, comprised of the following component tasks:

1. Find a direction in which to take-off.
2. Rise to an altitude where stable, manoeuvrable flight can be maintained.
3. Rotate around a fixed position and over several incremental steps.
4. Land back at base, "parking" in a home position.

The helicopter is controlled by a TIVA board that is programmable over a web-based remote laboratory interface (http://eng-labshare.canterbury.ac.nz); development work will be completed on TIVA-Orbit boards loaned in labs. The main project output will be a program for a TIVA board that contains a small, interrupt-driven, real-time kernel able to control the lift-off, hover, heading, and landing of a helicopter mounted on a remote rig.



```
Yaw: 12 [ 0]

Main 8 Tail 3

Mode: flying

----------------

Alt: 0 [ 0]
```

- Select button pressed.
- Finished loading demonstration program.
- Up button pressed.
- Up button pressed.
- Select button pressed.

Figure 1: The remote lab interface.

Physical Interface

The helicopter is attached to a rig that allows a limited vertical range of motion (altitude) and free rotation about the helicopter's z-axis (yaw).

The helicopter's rotors are driven by two motors, one for the main rotor and the other for the stabilising tail rotor. Both motors are controlled by pulse width modulation (PWM) outputs from the TIVA board and are powered by bench power supplies, which supply up to 6 A to the main rotor at full power. The main rotor rotates clockwise (viewed from above); the tail rotor generates a torque which opposes the torque of the main rotor.

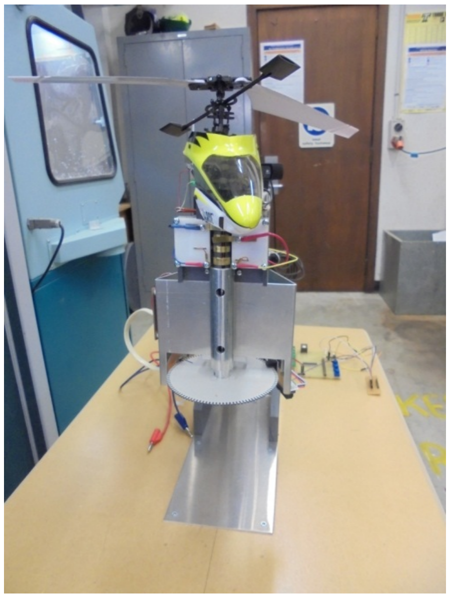


Figure 2: The HeliRig.

Feedback to the TIVA board is in the form of altitude, yaw, and reference orientation signals. Altitude is monitored by an optical distance transducer that produces an analogue signal between ~2 V (down) and ~1 V (up), corresponding to approximately 70 mm altitude range. Yaw is monitored by means of a quadrature optical angle encoder which generates two binary signals at voltage levels compatible with the TIVA board. There are 112 slots around the circumference of the angle encoder (i.e. 112 slots over 360°). Yaw movement isn't limited or restricted in any way; the helicopter can rotate freely. A reference signal is asserted (HIGH) when the helicopter faces directly towards the camera.

Program Specification

The developed program will be assessed against the following specifications:

1. The 2-channel quadrature signal for the helicopter yaw should be processed via interrupt-driven or polled decoding. **NB: There is a built-in quadrature encoder interface (QEI) on the TIVA board – this should NOT be used by the program.** The helicopter yaw in signed degrees should be displayed on the Orbit OLED display.

2. The analogue signal from the altitude sensor should be sampled and converted by the TIVA ADC (module 0, channel 9); the helicopter altitude should be displayed as a percentage on the Orbit OLED board. 100% should indicate maximum altitude, 0% should indicate minimum altitude, i.e. when the helicopter has landed. A reference minimum altitude reading should be recorded before take-off. To reduce the effects of electrical noise, readings should be filtered to reliably estimate altitude.

3. Two output PWM signals should be generated, one for the main rotor motor (module 0, PWM 7), and one for the tail rotor motor (module 1, PWM 5). The frequency of the PWM signals should be in the range from 150 Hz to 300 Hz. The PWM duty cycles for each motor should be displayed as percentages on the Orbit OLED display.

4. The SW1 slider switch on the Orbit board should control the mode of the helicopter: the DOWN position of the switch should indicate that the helicopter is either landed or in the process of

landing. *Changing* the slider switch from DOWN to UP when the helicopter is landed should cause the helicopter to take off.

- The helicopter should land smoothly, facing the reference orientation. When landing is complete the rotor motors should stop.
- At take off the helicopter should increase main rotor thrust until it is at the point of starting to hover.
- With the main rotor near the hover point, the helicopter should rotate to a known reference position, indicated by a HIGH input from the independent yaw reference signal.
- While the helicopter is landing any change to the switch position should be ignored until landing is complete.
- When the program starts, which may happen after a 'reset' operation (either physically on the TIVA or remotely over the web interface) or after reprogramming, the helicopter should be in landed mode with the motors off, regardless of the slider position. **NB: The web interface implements SW1 as a push button rather than a slider switch. This means that two pushes of the virtual button may be required to make the helicopter fly again after re-programming or a restart.**

5. The four user-programmable buttons of the TIVA-Orbit board should be programmed to interactively control the altitude and yaw of the helicopter.

    - Each operation of the UP button on the Orbit board should cause the helicopter to increase its altitude by 10% of its total vertical range. If the helicopter is already within 10% of its maximum altitude, pushing the UP button should take the helicopter to its maximum altitude. When the helicopter is at its maximum altitude any operation of the UP button should be ignored.
    - Each operation of the DOWN button on the Orbit board should cause the helicopter to decrease its altitude by 10% of its total vertical range. If the helicopter is already within 10% of its minimum altitude, pushing the DOWN button should take the helicopter to its minimum altitude. When the helicopter is at its minimum altitude any operation of the DOWN button should be ignored.
    - Neither the UP nor the DOWN button should affect the yaw of the helicopter.
    - Each operation of the CCW (left) button on the TIVA board should cause the helicopter to rotate 15° counterclockwise (viewed from above).
    - Each operation of the CW (right) button on the TIVA board should cause the helicopter to rotate 15° clockwise (viewed from above).
    - Neither the CCW nor the CW button should affect the height of the helicopter.

6. The actions of the slider switch and the four pushbuttons will be emulated by virtual signals which are sent to a remote helicopter rig over the remote lab interface. Therefore, the program should respond to pushbutton signals that are asserted for 200 ms for each "push"; only one button can be operated at a time. The virtual signal for the slider (MODE) is HIGH in the UP position.

7. Operation of the virtual RESET button (active LOW) should invoke a call to the API function `SysCtlReset()` (prototype in `driverlib/sysctl.h`).

8. The program should have a real-time foreground/background kernel, at minimum operating on a round-robin basis for background tasks. Robust behaviour should be maintained at all times.

9. At minimum a PI controller with a constant offset should be implemented to achieve controlled "flight".

10. Information on the status of the helicopter should be transmitted via a serial link from UART0 at 9600 baud, with 1 stop bit and no parity bit in each transmitted byte. Status information should include the desired and actual yaw (in degrees), the desired and actual altitude (as a percentage

of the maximum altitude), the duty cycle of each of the PWM signals controlling the rotors (%, with 0 meaning off) and the current operating mode. The information needs to be in a concise but easily readable format that is compatible with the scrolling display on the remote lab web page. Updates should be transmitted at regular intervals (no fewer than 4 updates per second).

11. The program should use the following pin map and on-board resources:

TIVA Pin	MCU Pin	I/O	Signal	Notes
J1-10	PA7	In	MODE	Slider switch (HIGH = up) & virtual signal
J2-03	PE0	In	UP	Active HIGH; virtual signal pulses HIGH on operation
J3-05	PD2	In	DOWN	Active HIGH; virtual signal pulses HIGH on operation
J4-10	PF4	In	CCW	Active LOW; virtual signal pulses LOW on operation
J2-04	PF0	In	CW	Active LOW; virtual signal pulses LOW on operation
J1-09	PA6	In	RESET	Active LOW; virtual signal pulses LOW on operation
J3-10	PF1	Out	Tail rotor motor	M1PWM5; 2% <= duty cycle <= 98 %, otherwise off
J1-03	PB0	In	Yaw channel A	Channel B leads channel A when yawing clockwise
J1-04	PB1	In	Yaw channel B	
J4-04	PC4	In	Yaw reference	LOW when helicopter is at the reference position, HIGH otherwise
J4-05	PC5	Out	Main rotor motor	M0PWM7; 2% <= duty cycle <= 98 %, otherwise off
J1-05	PE5	In	Altitude (analogue)	M0AIN9; approx 1 V - 2 V range, decreases with increasing altitude

Assesment

The project is worth 40% of your final grade for ENCE361. Marks for the project will be awarded for the whole group, but will factor in contribution statements. To maximise the chance of successful completion, the project is structured in terms of three sequential milestones, as follows:

Component	Due	Weight	Notes
Milestone 1	Week 5 (20-24/3)	7%	In-lab, requires code on git
Milestone 2	Week 8 (1-5/4)	5%	In-lab, requires code on git
Demonstration	Week 10 (15-19/5)	10%	In-lab, requires code on git
Code submission	6pm after your demonstration lab	10%	Marked off of eng-git Gitlab
Report and contribution statements	Friday of week 11 (26/5) 6pm	8%	Dropbox on Learn
TOTAL:		**40%**	

Grades for the milestones will be awarded for how closely the milestone specifications are met (see separate Milestone Specification documents). Grades for the demonstration will be awarded for how closely the specification is met. Grades for the code and report will be awarded according to the following rubrics:

Code	Marks
Style, layout and documentation	5
Modularity and structure	5
ISR choice and implementation	5
Scheduling of BG tasks	5
Total ÷ 2	**10%**

Report	Marks
Task identification & analysis	5
Schedular & inter-task comms	5
Design & justifications	5
Conslusions and overall report quality	5
Total ÷ 2	**10%**

Group Work

The project is designed to be completed by groups of two. Students can sign up to a group by following the relevant link on Learn. Students not allocated to a group of two after the online selection closes deadline will be randomly assigned. Group membership cannot be changed after the assessment of Milestone 1. One TIVA-Orbit board will be issued per group.

A git repository for each group will be created at https://eng-git.canterbury.ac.nz; you should commit code regularly to this repository. The portion of the project mark relating to source code will be based on code checked out from your repository at the relevant deadline. Remember to include your group name on all source files and on your project report.

Source Code

Source code from labs and lectures is available on Learn. TivaWare examples are also available from TI. You are welcome to reuse this or other code in your project on **three** conditions:

1. You explicitly acknowledge the provenance of any code you use, typically by way of comment in your source files.
2. You take responsibility for any code you use: this responsibility includes changing the comments to reflect your use of the code.
3. You will not receive marks for code you did not write.

You will lose marks if you do not meet these conditions.

Serial Port

During development groups may choose to output debugging information over the serial port, in addition to the status information required by the specification. This debugging information should not be included in the final version of the code. You are strongly encouraged to use compiler directives to control the transmission of such debug information. For example, a section of code which should not be compiled can be bracketed as follows:

```
#define DEBUG // Line can be commented in/out to toggle debug.
#ifdef DEBUG
    // Code for debugging here; only compiled if DEBUG is #defined
#endif
```

Learning Support and Troubleshooting

Tutors and lecturers will be available during regular lab sessions throughout term 1 and 2. Additional support is available via the Learn forum, where students can ask (and answer) questions related to the course.