Flight Control System Description and Specifications

The autopilot for the flight test vehicles consists of the Cloud Cap Technology Piccolo II Autopilot. The commercial off-the-shelf Piccolo autopilot system for rotorcraft provides the user with an immediate capability to establish fully automated flight operations, including take-off, waypoint following, and landing, with minimum dependence on dynamic modeling. The flight portion of the system consists of the proven Piccolo hardware set with integrated microprocessor, sensor suite, and spread spectrum serial radio transceiver.

The processor employed is the Motorola 555. The internal sensor suite consists of a GPS receiver and a three-axes set of both rate sensors and accelerometers to produce a GPS-aided inertial navigation solution. Also available are a set of pressure transducers for determining altitude and airspeed, but these are not employed on a standard helicopter. The rotorcraft solution requires an external 3 axes magnetometer to determine magnetic heading, and sensing of rotor or engine RPM. For automatic take-off and landing, the system also normally employs an external laser altimeter to measure height above ground. The flight system requires a GPS antenna, and a data link antenna.

The system can be piloted manually via a standard joystick interface, or can be tasked by a ground station operator to perform in a fully automated manner. The communication of pilot commands, as well as commands from the operator interface, and all telemetry from the vehicle to the ground are managed using a single spread-spectrum serial radio link (operating on the range 902-928 MHz).

The ground hardware employed includes a small suitcase referred to as the ground control station or GCS. The GCS has its own microprocessor dedicated to management of communications with the vehicle. A standard hobby-style hand-held joystick interface for remote control piloting is attached to the GCS with a “trainer” cable. The GCS also has a GPS receiver so that the location of the GCS can be properly located on map displays. When selected as a specific hardware option, this receiver can also be used to provide the vehicle with differential GPS corrections. The GCS normally runs from a 120 V AC power input, or a 12V DC input, and has an internal battery backup system. The GCS also requires a GPS antenna, and a data link antenna.

The system is managed by an operator with a laptop computer connected serially to the GCS and running application software known as the Piccolo Command Center. This software enables the operator to plan and store missions, to configure the vehicle sensors, actuators and control system, to monitor mission execution and system health, to display the vehicle state on various map and photo overlays, and to store the telemetry data to disk.

Advantages of the GCS design include the ability to manage multiple vehicles from a single ground station, including the ability to manually pilot a selected vehicle while the others are automated. Also, because the communications is managed by the GCS, and not the application software employed by the operator on a PC, the PC, which may also be running other application software, can become corrupted and be rebooted without disrupting the flight operation. It is also possible in this design for multiple copies of the Command Center to be running on multiple PCs, all networked together, and for all of them to share vehicle data and for each to issue commands to the vehicles.

The control system consists of a set of adaptive inner loops for tracking attitude commands. Adaptive outer loops for tracking position and velocity commands generate the attitude commands. The loops are designed using model inverting control and very simple models, and are augmented with neural networks that learn in real-time to compensate for any model inversion error. This produces a system that is extremely robust to parameter uncertainty, and essentially eliminates the traditional dependence of control system design on wind-tunnel testing and gain scheduling. The position and velocity commands are generated from a set of rules that are implemented via a state machine to create a “command generator”. The mission is prescribed in the form of waypoints. These can be points where the vehicle is commanded to hover, as well as “fly through” waypoints, or points where hover is performed until a timer expires. One can command heading directly, or slave heading to track the velocity vector in the horizontal plane. Take-off and landing waypoints have their own state machines, and are completed by sequentially executing a set of relevant tasks. There is also a “lost communications waypoint” that the user employs to prescribe the events that should occur in the event communication with the vehicle is lost for more than a specified period of time. If this waypoint is a “landing” waypoint, the vehicle will execute a fully automated landing once it has returned to the prescribed location.

One can prescribe a number of distinct missions, which are stored on board. At any time the operator can “point and click” to reassign the next waypoint, which may be in an alternate mission plan, or may insert or drag waypoints to alter the current mission plan during execution. One can also suspend mission execution at any time, and the vehicle will simply hover at its current location. One can then issue “steering” commands to “push” the vehicle around (i.e. up or down, forward or backward, and so forth) while fully stabilized in order to conduct exploration of an area without a specific mission plan, or to support interactive placement of a sensor. Selecting a waypoint while in this mode will then cause the vehicle to return to execution of a normal mission waypoint sequence.

Piccolo II Specifications:

- Power: 8 V to 20 VDC, 5 W max
- Size: 5.6 inches x 3.0 inches x 2.4 inches
- Weight: 233 grams (8.2 oz)
- Radio: 900 MHz Unlicensed ISM
- GPS: 4 Hz
- Serial Interface: Five RS232 payload ports
- Servo PWM: Up to 16 servos
- CAN Bus: Simulation Interface
- General Purpose: 6 GPIO + 4 analog inputs