

GSM Technology as a Communication Media for an Autonomous Unmanned Aerial Vehicle

Mariusz Wzorek, David Landén, Patrick Doherty
Department of Computer and Information Science
Linköping University, SE-58183 Linköping, Sweden
{marwz,davla,patdo}@ida.liu.se

ABSTRACT

Unmanned Aerial Vehicles (UAVs) are becoming more reliable, autonomous and easier to use with great potential for commercial use in common airspace in the near future. Though autonomous UAVs often do not rely on communication links with the ground during flight, communication is necessary and essential in accomplishing complex mission tasks where the ground operator is an essential part of the mission. Communication links become especially important in cooperative missions where tasks are solved using many vehicles of different sizes and characteristics (ground or airbourne) with different sensor suites.

Data links, such as wireless Ethernet or radio modems that use open frequency bands are often unreliable in urban areas due to interference from other users, spreading and reflections from terrain and buildings, etc. GSM and its related technologies GPRS, EDGE, 3GSM offer an interesting communications infrastructure for remotely accessing, controlling and interacting with UAVs in an integrated and highly portable manner and offer the ability to interface to the WWW for additional information useful in mission achievement.

This paper describes a case study of feasibility of using such technologies for UAV operations. A prototype network is presented, that was created between two UAVs and a ground operator using GPRS technology. Experimental results describe performance and reliability of the network. A graphical user interface for a Sony Ericsson P900 mobile device was designed and implemented using Java. It provides the ground operator with a portable control interface for a UAV and its camera. It also receives telemetry data (e.g. position, altitude, state, etc.) from the UAV in addition to an image stream from the camera. The results show that GSM network infrastructures provide a useful means for communicating with UAVs, especially in urban areas. The proposed solution can be used as a complementary data link to improve robustness, reliability and range of the communication channel. These techniques have been tested in actual flight.

BIOGRAPHY

Patrick Doherty is a Professor at the Department of Computer and Information Science (IDA), Linköping University (LiU), Sweden. He is director of the Artificial Intelligence and Integrated Computer Systems Division at IDA and his research interests are in the area of knowledge representation, automated planning, autonomous systems, approximate reasoning and UAV technologies. He is also program director of LINKLAB, a center for future aviation systems. LINKLAB is a joint endeavor between LiU and Saab Aerosystems, Sweden.

Mariusz Wzorek is a graduate student at the Department of Computer and Information Science (IDA), Linköping University (LiU). He received his MSc degree in engineering from University of Technology and Agriculture in Poland and MSc degree in computer science and engineering from Linköping University in Sweden. Main focus of his research is automated planning techniques, autonomous unmanned systems and robotics.

David Landén is a graduate student at the Department of Computer and Information Science, Linköping University. He received his MSc degree in computer science and engineering from Linköping University, Sweden.

1 Introduction

Unmanned Aerial Vehicles (UAVs) are rapidly becoming a key technology in the military domain and offer great promise as a useful technology in many commercial and civil applications in the future. The potential applications are highly diverse and the associated requirements on platform type and payload, in addition to ground control stations and interfaces, must match the constraints of the application in order to make their use both user-friendly and economically feasible.

There are many different types of UAVs ranging in size and weight from Predator type HALE systems with wingspans of 14.8m to Micro Air Vehicles such as the LinkMav ¹ with rotor diameter under 49cm. Whereas the Predator requires a highly sophisticated ground control station with a large number of operators, UAVs in the micro- and mini- class are often intended to be used on-the-fly out in the field, with little pre-flight preparation time, and with highly flexible deployment characteristics. In the latter case, it is important to develop ground control stations and interfaces to the UAV in as lightweight and modular a manner as possible. Leveraging existing COTS technology is also important in order to help make such systems commercially feasible.

The Autonomous UAV Technologies Laboratory ² at Linköping University, Sweden, has been developing fully autonomous rotor-based UAV systems in the mini- and micro-UAV class. Much effort has gone into the development of useful ground control station interfaces which encourage the idea of *push-button missions*, letting the system itself plan and execute complex missions with as little effort as possible required from the ground operator other than stating mission goals at a high-level of abstraction and monitoring the execution of the ensuing mission. When viewed from this perspective, an obvious line of research is to develop highly portable, lightweight ground control station systems such as mobile telephones or PDAs.

An example of such a push-button mission that has been used as an application scenario in our research is a combined monitoring/surveillance and photogrammetry mission out in the field in an urban area with the goal of investigating facades of building structures and gathering both video sequences and photographs of building facades. For this experiment,

we have used a Yamaha RMAX helicopter system as a platform. Let's assume the operational environment is in an urban area with a complex configuration of building and road structures. A number of these physical structures are of interest since one has previously observed suspicious behavior and suspects the possibility of terrorist activity. The goal of the mission is to investigate a number of these buildings and acquire video and photos from each of the building's facades. It is assumed the UAV has a 3D model of the area and a GIS with building and road structure information on-line.

The ground operator would simply mark building structures of interest on a map display and press a button to generate a complete multi-segment mission that flies to each building, moves to waypoints to view each facade, positions the camera accordingly and begins to relay video and/or photographs. The motion plans generated are also guaranteed to be collision-free from static obstacles. If the ground operator is satisfied with the generated mission, he or she simply clicks a confirm button and the mission begins. During the mission, the ground operator has the possibility of suspending the mission to take a closer look at interesting facades of buildings, perhaps taking a closer look into windows or openings and then continuing the mission. This mission has been successfully executed robustly and repeatedly from take-off to landing using the RMAX.

One of the research goals has been to take such application scenarios out in the field where one assumes the ground operator should be equipped with as portable an interface to the UAV as possible, making his/her movements more flexible and stealth-like in such operations. A lighter weight version of this mission scenario has been successfully executed using a Sony Ericsson P900 as ground control station and interface to the RMAX. Details will be described in the paper. One can envision many other mission scenarios and applications where lightweight, portable interfaces are not only useful, but perhaps a requirement. Human resources in such missions should be able to call UAVs and interact with them on the telephone, much like one would do with human to human collaboration. We view this experimentation as a first step in achieving that longer term goal.

There are a great many advantages to using mobile telephone technology in this context.

- **Ground Control Portability and Mobility** – Many mission scenarios and applications will require out of the backpack, power efficient, battery-driven, solutions, especially when the

¹www.linklab.se

²www.ida.liu.se/~patdo/auttek/

micro air vehicle platforms mature. Mission scenarios will be specified at a high-level of abstraction, taking advantage of the increasing degree of autonomy in such systems. This perspective minimizes bandwidth requirements and the need for complex ground control solutions and interfaces, while still keeping the ground operator in the loop. Here one is interested in uplinking high-level commands and downlinking simple types of telemetry and mission progress feedback.

- **Leveraging COTS Technology** – 3G technology is developing at a high rate of acceleration. New functionalities in future generations of mobile telephones can be used to advantage in the UAV domain. Some examples of this are Google Maps, GPS capability, and WWW access to databases and other information on-the-fly, in addition to increased bandwidth and technology for downloading video streams in real-time and guaranteeing quality-of-service due to the large consumer base. Mobile telephone technology is also in the process of being merged with PDAs and laptops offering potentially more powerful computational capabilities on the ground.
- **Cost Efficient Solutions** – GSM and future mobile telephone technology offer extended mission range at a low cost by taking advantage of an existing GSM infra-structure rather than higher-priced RF solutions or limited range wireless Ethernet solutions. This provides the potential for out-of-sight flying with low infra-structure investment. In addition, it offers the prospect of assisting in the development of indoor navigation, in particular in the context of MAVs.

GSM and Mobile telephone technology are mature enough to begin experimentation and research into their integration with UAVs and their use in complex mission scenarios, even though there are still problems with coverage, bandwidth, robustness and reliability in the current generation of systems. This paper describes our current efforts in this direction.

1.1 Paper Outline

The paper is structured as follows. In section 2, we begin with an overview of existing and forthcoming mobile technology and defend our design decisions based on this analysis. In section 3, we describe both

the UAV platform, the autonomous system integrated with that platform and the GSM UAV communication network which has been developed and used in the experimentation. In section 4, we describe experimental results regarding datalink performance measurements in the GSM UAV communication network. In section 5, we describe the multi-modal user interface developed on a Sony Ericsson P900 mobile telephone used to setup UAV missions in an urban area. In section 6, we describe actual flight tests which successfully use the GSM-based communication network in a number of complex scenarios where the RMAX has been deployed. In section 7, we conclude and discuss future work.

2 Mobile Technologies

The choice of a communication technology used with UAVs strongly depends on the application area. In the UAV domain there are many applications with different requirements on the datalink performance between ground operator and UAV or between UAVs themselves. In this section, we describe some of these requirements relative to application, provide an overview of GSM related technologies that are candidates for use in our communication network, and conclude with the design decisions made based on this analysis.

Data links can be specified in terms of latency, bit rate and data package loss rate. Different applications place certain constraints on these parameters. The first type of datalink connection of interest is the link between one or more ground operators and one or more UAVs. The degree of autonomy assumed for a UAV greatly influences the latency, bit rate and package loss rate constraints. Very little autonomy places much greater demands than communication with a fully autonomous UAV.

In section 1, we described the concept of pushbutton missions. In this case, user interfaces designed to help control UAVs generally do not require high bandwidth uplinks since only high-level goals or action commands are sent to the UAV rather than large sequences of control signals to control the platform itself. The types of commands uplinked are sporadic and of the type, fly-to-point, take-off, hover-at-point, land, etc., while mission goals might be of the type, fly-to building and monitor the front of the building. In these cases, there is very little interaction between the ground operator and UAV, and consequently weaker constraints on latency,

bit rate and data package loss rate.

The bandwidth requirements for the downlink between a UAV and ground operator again are dependent on the level of autonomy assumed for the UAV and on the amount of data the ground operator requires to complete a mission task. For an autonomous UAV, often one is only concerned with receiving situation parameters such as the current command status, position of the UAV, its altitude, etc. In this case low bit rate requirements suffice. The bandwidth requirements increase relative to type of sensor and with the amount of sensor data requested by a ground operator. For instance, if a UAV is equipped with a video camera, streaming of images would require much higher bandwidth than for the case where a low-resolution photo is required every few seconds. The latency in this type of connection is not crucial unless real-time data streaming is required.

The second type of connection of interest is that between two or more UAV platforms. Mission configurations are rapidly moving from a single UAV and ground operator to multiple UAV platforms with several ground operators. In this case, the intent is that UAVs cooperate with each other to achieve mission tasks based on the resources each UAV has at its disposal. In this case, distributed problem solving techniques are used to achieve mission tasks. Such tasks often include resource or information sharing depending on the degree of centralization in the solution. It is assumed that different platforms will have different capabilities and different sensor suites. In order to solve a specific mission, one UAV may require the use of another platform's resources.

For instance, one platform may have more computation power and more sophisticated deliberative services (e.g. planners) that other platforms may not have. In this case the other platforms can use those resources to help in achieving the mission goals delegated to them. A good example of information sharing is that of decentralised simultaneous localization and mapping (11), where a fleet of UAVs builds local maps independently and then merges them into one coherent global map. There are many such scenarios and the requirements on the bandwidth and the latency of the link can vary from very low to very high. At a lower level of abstraction, there is much interest in cooperative control and formation flight among a squad of UAVs. Formation flight requires tight synchronization between UAVs, thus the delay in communication links has to be considerably small. For formation flight applications the bandwidth is of less importance.

All of the requirements placed on the data links that we mentioned can be satisfied by using standard technologies such as wireless Ethernet or radio modems. In some cases, e.g. data links for out-of-sight communication, it would be very expensive though. Mobile technologies such as GSM networks offer an interesting alternative. The infrastructure for such networks is already built and standardized in many countries. The connection costs are relatively low, although the services are not always reliable. Additionally, GSM technologies open up yet another possibility for UAVs and even ground operators to extend their knowledge bases and computational power by accessing resources directly on the Internet.

Although mobile communication technology became commercially available in the early 80s when the first generation (1G) (12) analog cellphone standards were created, the real boom started with the introduction of the digital mobile communication (2G) in middle 90s. The Global System for Mobile Communications (GSM) is the most popular standard for mobile phones in the world. GSM services are used by over 1.5 billion people across more than 210 countries and territories. GSM uses Time Division Multiplexing (TDM). It is a method of putting multiple data streams in a single signal by separating the signal into many segments, each having a very short duration. Each individual data stream is reassembled at the receiving end based on the timing. In the next subsections, we present more detailed descriptions of available wireless mobile technologies for data transmissions.

2.1 HSCSD

High Speed Circuit Switched Data (HSCSD) is an extension of the Circuit Switched Data (CSD) data transmission mechanism. In the CSD, channel allocation is done in circuit switched mode where a dedicated connection (circuit) is set up for the whole transmission duration, such as with a normal phone call. CSD uses a single radio time slot to deliver 9.6 kbps throughput.

In the HSCSD more than one time slot can be used during a connection increasing the total data throughput (as we would make many phone calls in parallel). The connection costs are calculated based on the duration of the connection and the number of slots that were used. In theory, during one session, up to six time slots can be allocated which gives the maximum theoretical data rate 86.4 kbps. In practice, it can never be obtained, since the A-interface (between the base station controllers

and the mobile service switching center) in the GSM network is built on ISDN technology with 64 kbps channels (6). The maximum throughput for the devices with four slots support is presented in Table 1.

In the case of many users competing for the time slot allocation, the highest priority have voice calls. Even if a connection has already been established over many time slots it is not guaranteed that the time slots will not be redistributed to the new awaiting voice calls when the network is overloaded.

Slot configuration	Download rate [kbps]	Upload rate [kbps]
2 slots for download 2 slots for upload (2+2)	28.2	28.8
3 slots for download 1 slot for upload (3+1)	43.2	14.4

Table 1: Maximum data transfer rates for chosen configurations using HSCSD.

2.2 GPRS

The General Packet Radio Service (GPRS) is a data transmission technology built on the GSM foundation. It gained popularity due to relatively low costs for a connection and availability over vast areas. In GPRS, a Gaussian Minimum-Shift Key (GMSK) modulation technique is used to code 1 bit of information per pulse rate. GPRS uses the same time frame with eight time slots as GSM voice transmission. A single time slot is called a packet data channel (PDCH). The allocation of the PDCHs is dynamic and the number of slots varies in time depending on the amount of transmitted data. Compared to the circuit switched GSM (HSCSD or CSD), where a time slot is held for the time of the session or the call, creating a static circuit (route), GPRS is packet driven and each packet can travel a different route during one session.

Depending on the communication conditions, one of four different coding schemes is used: CS1 to CS4 with varying levels of forward error correction (FEC). The mobile station can change coding scheme dynamically depending on the changes of the communication conditions. The CS1 coding scheme uses half of the bits in the radio block for error correction and detection. On the other hand, the CS4 has no error handling at all and only the payload is transmitted. Most of the currently

deployed mobile networks support only CS1 and CS2 coding schemes. This is because the error rates for CS3 and CS4 coding schemes would be typically too high.

The coding scheme and the number of allocated slots determine the maximum data rate for the GPRS communication. In the definition of the GPRS standard, a single user can allocate up to eight time slots. In practice, it is impossible due to several reasons. One of them is the limitation of the mobile terminals that usually support up to five slots at most, due to limited energy resources. Another reason is that most mobile operators hold higher priority for telephone calls and circuit switching data than for GPRS. Such a restriction is most likely made for economical reasons, since the operators get most of their revenues from voice calls.

The theoretical maximum transfer rate in GPRS is 22kbps per time slot (assuming CS4 coding scheme). This implies data transfer rates up to 170 kbps when using all eight slots. However, in practice, due to limitations on time slots usage, lower priority of GPRS data and coding schemes, the maximum data transfer rates are much lower. Table 2 presents the data transfer rates for the case of using 5 time slots and CS1, CS2 coding schemes. The round-trip time inside the GPRS network is 800 ms (13).

	CS1 coding scheme	CS2 coding scheme
(4+1) ¹	36.2 kbps download 9.05kbps upload	53.6 kbps download 13.4 kbps upload
(3+2) ²	27.15 kbps download 18.1 kbps upload	40.2 kbps download 26.8 kbps upload

Table 2: Maximum data transfer rates for chosen configurations in GPRS network. ¹- 4 slots for download, 1 slot for upload; ²- 3 slots for download, 2 slot for upload

2.3 EDGE

Enhanced Data rates for Global Evolution (EDGE) is the successor of GPRS. EDGE uses Linear 8-Phase Shift Key (8-PSK) modulation which is more bandwidth efficient than the GMSK modulation used in the GPRS standard. The two phase positions in GPRS, have been replaced by eight positions in EDGE. This means that instead of sending one bit per each symbol, it is possible to send three bits per each symbol increasing the data rate by a magnitude of three. The downside of such a modulation scheme is greater sensitivity for

interference, since the phase positions are closer to each other than in GPRS. A symbol that is slightly altered might be mistaken for another of the 8-Phase symbols.

EDGE adds six more coding schemes to CS1-CS4 for GPRS, with a different number of bits used for error handling. The part of a radio block used for actually sending user payload varies from 33% for PCS1 to 100% for PCS-6 (with no error correction). The last one provides the bit rates of 69.6 kbps. This provides a high theoretical data rate for EDGE compared to GPRS. But just as for GPRS, there is no guarantee of acquiring more than one slot nor that the highest coding scheme can be used all the time. The typical data rates will be about 120-160 kbps for 3-4 downloading slots using coding schemes PCS2-PCS3. The round trip time, 400 ms, is improved compared to GPRS.

EDGE uses the same GSM infrastructure as GPRS but requires an upgrade of base station hardware and at the time of writing this article it is not yet widely used in Europe. However, many mobile operators have already announced introduction of the EDGE technology or offer it in the vicinities of big cities.

2.4 UMTS

The Universal Mobile Telecommunications System (UMTS) is the third generation (3G) variant for GSM. Unlike EDGE and GPRS, this technology is not built on Time Division Multiple Access (TDMA). Instead of dividing the frequency band into smaller parts and then dividing the usage of frequencies over time in frames and time slots, all users use the entire frequency band. In order to know which package is aimed at a certain user, each user gets an access code. The mobile station uses the access code to filter out messages addressed to certain users and everything else is regarded as noise. Such a technique is called a Wideband Code Division Multiple Access (W-CDMA).

This approach extends the capacity of the network (more users can access the network at the same time) and increases the data transfer rates. The download data rate for UMTS is 384 kbps, which is about twice the speed one could assume to obtain when using EDGE. Delay reduction is further improved and the round trip time is down to 90-150 ms.

UMTS is the newest technology and at this time only available in some countries in the vicinities of big cities.

2.5 Design decision

An optimal specification for data links in our UAV applications would include a low latency, a high data transfer rate and a large range (coverage). Comparing the technologies presented in the previous subsections, UMTS and EDGE have a definite advantage over others when considering the latency and the bandwidth. On the other hand, the lack of coverage outside the cities makes it unusable for UAV applications for the time being. In the near future we can expect mobile operators to increase the coverage of UMTS and EDGE networks.

At the time of writing this article the most mature mobile technologies for data transfer are HSCSD and GPRS. For the design of our data link we have chosen GPRS technology. Although HSCSD offers lower latencies for connections, GPRS has yet more to offer. One of the main advantages of GPRS technology is the relatively low cost for each connection. The fee for a connection is based on the amount of data sent and not, as in the case for HSCSD, on the duration of the connection. GPRS also offers higher throughput and shorter connection startup time. Figure 1 presents the network configuration developed and used in our experimentation.

3 UAV System Description

In this section we present a description of the UAV platforms used in the experiments. In the first subsection we provide a description of the hardware including the specification of the GPRS wireless modems. In the second subsection, we give a short overview of the software system.

3.1 The Hardware Platform



Figure 2: The WITAS RMAX Helicopter

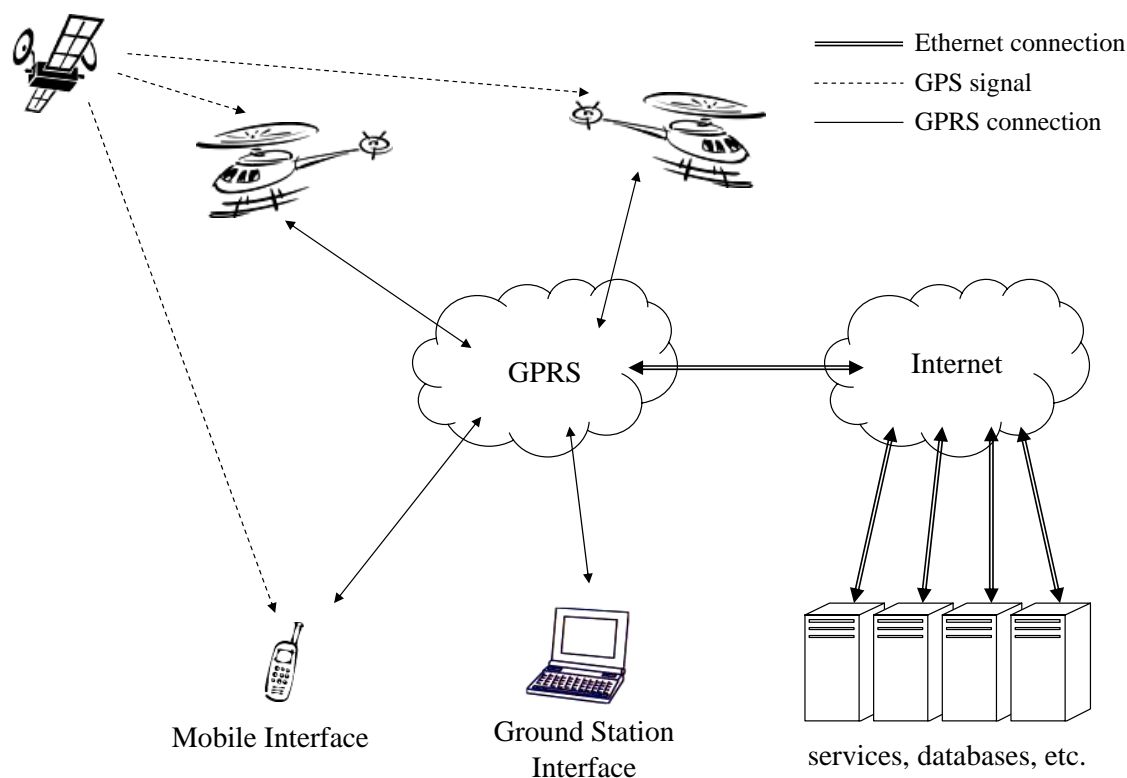


Figure 1: Communication network overview

The WITAS³ UAV platform (5) is a slightly modified Yamaha RMAX helicopter (Fig. 2). It has a total length of 3.6 m (including main rotor) and is powered by a 21 hp two-stroke engine with a maximum takeoff weight of 95 kg. The helicopter has a built-in attitude sensor (YAS) and an attitude control system (YACS). The hardware platform developed during the WITAS UAV project is integrated with the Yamaha platform as shown in Fig. 3. It contains three PC104 embedded computers. The primary flight control (PFC) system runs on a PIII (700Mhz), and includes a wireless Ethernet bridge, a GPS receiver, and several additional sensors including a barometric altitude sensor. The PFC is connected to the YAS and YACS, an image processing computer and a computer for deliberative capabilities. The image processing (IPC) system runs on the second PC104 embedded computer (PIII 700MHz), and includes a color CCD camera mounted on a pan/tilt unit, a video transmitter and a recorder (miniDV). The deliberative/reactive (DRC) system runs on the third PC104 embedded computer

³WITAS is an acronym for the Wallenberg Information Technology and Autonomous Systems Lab which hosted a long term UAV research project (1997-2004).

(Pentium-M 1.4GHz) and executes all high-end autonomous functionality. Network communication between computers is physically realized with serial line RS232C and Ethernet. Ethernet is mainly used for CORBA applications (see below) and remote login and file transfer, while serial lines are used for hard real-time networking. We have tested two wireless modems during experiments. They are connected to the DRC computer:

- Wireless EDGE/GPRS modem (Aplicom 12).
The modem supports GPRS class 10 (meaning 4+1 timeslots in total, configured as 4+1: receiving 38.4-53.6 kbps and sending 9.6-13.4 kbps, or as 3+2: receiving 28.8-40.2 kbps and sending 19.2-26.8 kbps). It also has HSCSD class 6 and EDGE class 6 support. It has an embedded TCP/IP stack, and provides the possibility to run small J2ME applications, called IMplets (like MIDlets but without the screen classes).
- Wireless GSM/GPRS modem (Trizium).
It has a specification similar to the Aplicom 12 modem (GPRS class 10, TCP/IP stack) but lacks HSCSD and EDGE support.

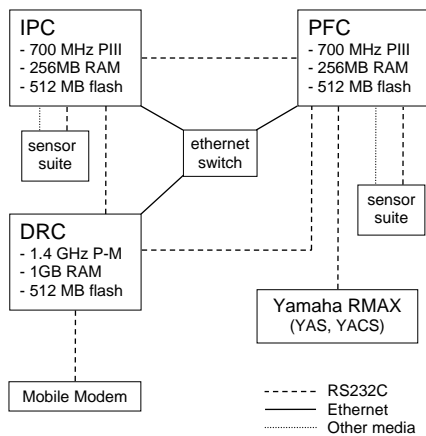


Figure 3: On-Board Hardware Schematic

3.2 The Software Platform

A hybrid deliberative/reactive software architecture has been developed for our UAV platform and has also been used in a ground robot. Conceptually, it is a layered system with deliberative, reactive and control components. The architecture has a *reactive concentric* flavor where reactive task procedures use services provided by both deliberative and control components in a highly distributed and concurrent manner.

The software implementation is based on CORBA (Common Object Request Broker Architecture), which is often used as middleware for object-based distributed systems. It enables different objects or components to communicate with each other regardless of the programming languages in which they are written, their location on different processors or the operating systems they running on. A component can act as a client, a server or as both.

The functional interfaces to components are specified via the use of IDL (Interface Definition Language). The majority of the functionalities which are part of the architecture can be viewed as CORBA objects or collections of objects, where the communication infrastructure is provided by CORBA facilities and other services such as real-time and standard event channels. This architectural choice provides us with an ideal development environment and versatile run-time system with built-in scalability, modularity, software relocatability on various hardware configurations, performance (real-time event channels and schedulers), and support for plug-and-play software modules.

Fig. 4 presents some (not all) of the high-level services

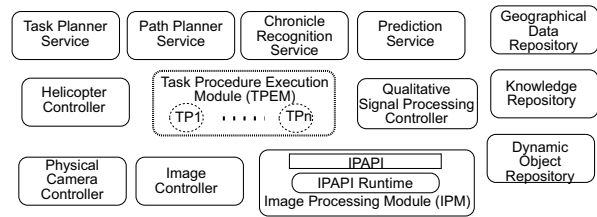


Figure 4: Some deliberative, reactive and control services

used in the WITAS UAV system. These services run on the DRC computer and interact with the control system.

The control system is a hybrid distributed system that runs primarily on the PFC computer in a real-time environment (7) constructed especially to integrate seamlessly with the rest of the architecture. Hierarchical concurrent state machines (HCSMs) are used to represent system states. The ability to switch modes contingently is a fundamental functionality in the architecture and can be programmed into the task procedures associated with the reactive component in the architecture. We have developed and tested several autonomous flight control modes: take-off, landing via visual navigation, hovering, dynamic path following, and reactive flight modes for tracking and interception. A CORBA interface is setup on top of the control system kernel so high-level components can issue commands to initiate and sequentialize different flight modes. Helicopter states and events from the control system are in turn sent to the high-level system.

The architecture design and the use of CORBA allows us to locate in flexible manner deliberative services on different computers. We used this feature extensively during flight test verification of the mobile data link.

4 Data Link Performance Measurements

Many studies have been made which investigate the performance of GPRS connections for different protocols (3; 4; 2). Available studies mostly focus on the TCP/IP protocol and show that there are real performance issues when using this protocol in mobile telephone networks. For that reason, we decided to base our communication protocols on the UDP protocol instead. In order to design a communication protocol that handles data loss, data out of order and delays, the parameters of the link have to be known. There

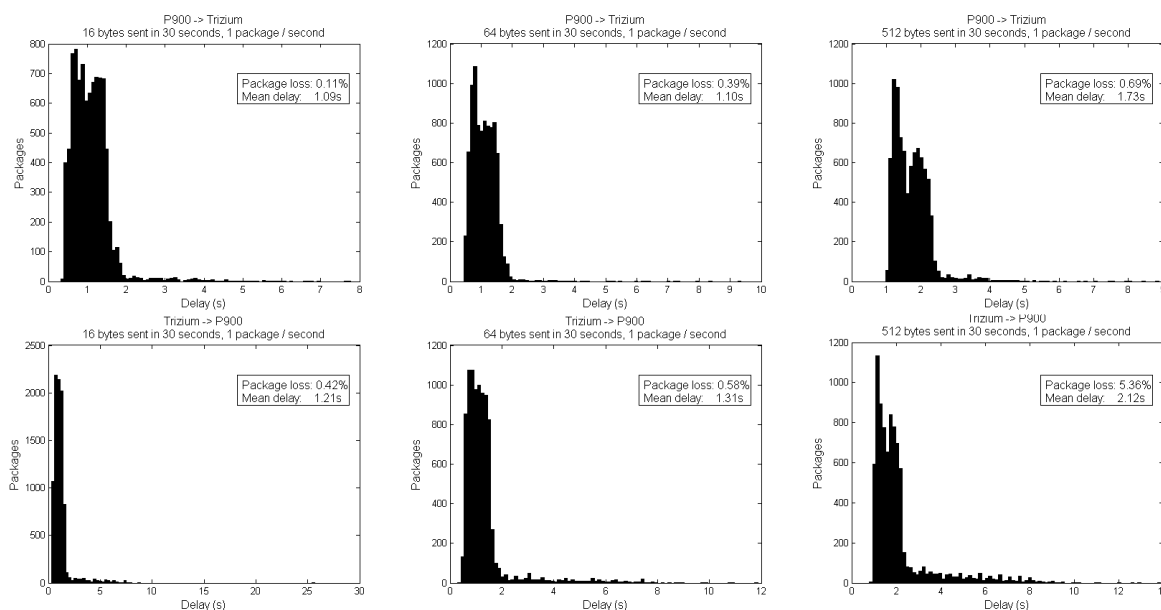


Figure 5: Test results for the connection between P900 and Trinzum devices

are many factors, e.g. mobile network infrastructure, quality of provided services etc., that can influence performance. Due to the many unknown factors it is hard to draw theoretical guarantees about the behavior of a particular network. Consequently, an empirical approach is necessary and we did this for our network.

The main goal of the tests which were conducted was to measure the frequency of package loss and delays. Tests were performed indoor and outdoor at different locations and different times of the day to provide reliable and generalizable results. A mobile phone was installed as a modem using point-to-point serial connection. Two other modems were connected to our flight computers as was described in section 3.1. Since one of the measured values was one-way-delay, the computer clocks had to be synchronized. An NTP protocol (9) was used for the synchronization.

The synchronization error was very small compared to the measured value so it could be neglected. UDP datagrams were used as a data. Every set of experiments included sending packets with the same size and at the same rate. Data traffic was generated and measured using the Distributed Internet Traffic Generator (D-ITG) (1). D-ITG can generate traffic on three different levels: the network, transport and application levels. Some of the supported protocols include TCP, UDP, ICMP, Telnet and VoIP. The traffic generator logs the departure and arrival time of the packets and derives the total time for the transfer,

minimum and maximum delay, average packet rate, average bit rate and the number of dropped packages.

In a preliminary set of tests, the performance of the link between a Sony Ericsson P900 mobile telephone and a Trinzium modem was measured. During those tests different package sizes were used i.e. 16, 32, 64, 128, 256 and 512 bytes. The sending rate varied between 1, 5 and 10 packages per second. Each combination was repeated 10 times. The results were used to choose interesting test cases for the next experiments. In the final set of tests, packages with sizes i.e. 16, 64 and 512 bytes were used. In each test case, the data was sent with a 1 package per second rate during 30 second periods. In each test suite between two devices, the tests were carried out in both directions. Finally, to assure a fair evaluation, we made 300 repetitions of each flow and split this number into three parts, each part run on a different day. Some of the results are presented in Figure 5 and 6.

General observations for all the three test suites were:

- no reordering of packages occurred
- when packages got lost, they were often lost in big bunches
- sometimes the delay value was high at the beginning of the transmission, and then stabilized
- the delay was significantly smaller for 16 and 64 bytes packages compared to 512 bytes packages

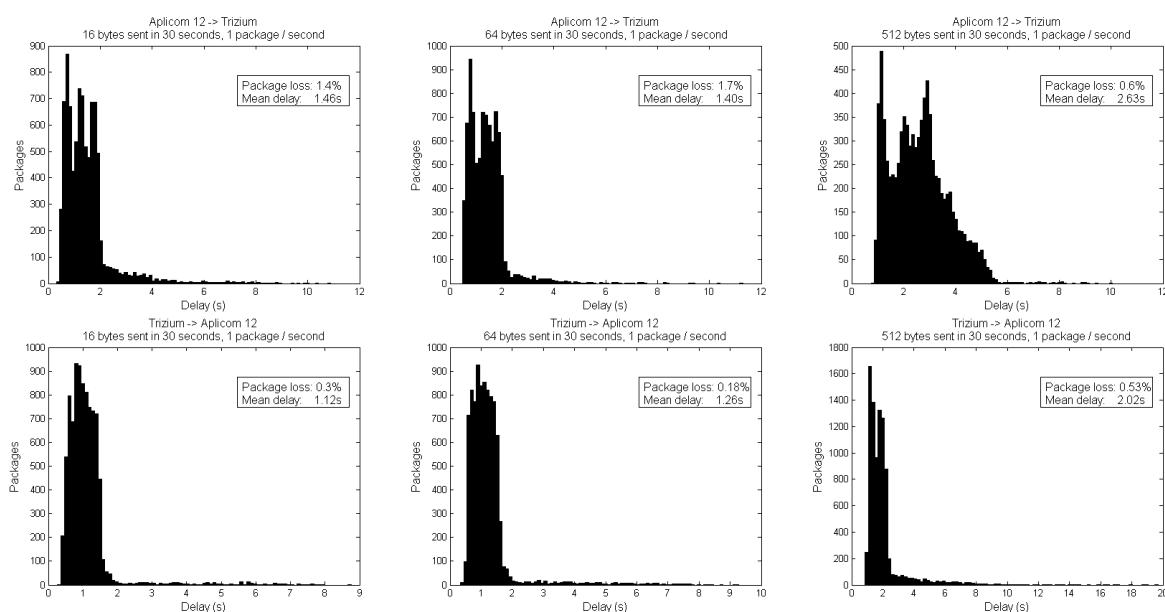


Figure 6: Test results for the connection between Aplicom 12 and Trinzum devices

- in most cases the data package loss was small, between 0.11%-1.7%, regardless of the package size. In one case the package loss was as high as 5.36%.
- the performance between the mobile phone and the two modems was very similar, as expected.

5 Mobile User Interface

In recent years, mobile technology has become an integral part of our everyday lives. Recent growth in this market sector has resulted in an increase in investments into mobile phone design resulting in many enhancements to what were previously viewed as just voice communication devices.

The new generation of mobile phones, often called smartphones, offer more features and computational power than a standard personal computer from the early 90s. The smartphones available on the market today are capable of running graphical user interfaces that can be used for remote control and interaction with UAVs.

Some of the advantages over conventional ground station control interfaces which often require a motor vehicle are their small size, low weight and long endurance (with small battery packs). Of course, mobile devices have some disadvantages such as limited resources (e.g. small screen size, small memory size,

etc.) and not all of the features of a stationary ground control station can be implemented on such devices. Because of these limitations, the design of a graphical user interface (GUI) for a mobile device is a challenging task.

We have chosen Java Micro Edition (J2ME) (8) as a programming environment for the implementation since it is supported by most of the commercially available smartphones. The graphical user interface contains two main views called the Map View (Figure 7) and the Camera View (Figure 8). The first one displays a two-dimensional map for the current operational area.

The map can be changed at the start of the application. In the Map View it is possible to zoom in and out, as well as move the map by dragging it with the stylus. Currently the maps are stored together with the application, but in future versions of the system, it will be possible to upload the map dynamically from the Internet or from a memory storage offered by the device (e.g. memory stick).

Commands are accessible in the four drop down menus at the top of the screen (Figure 7). The first head menu option (labeled **Map**) contains commands related to the current map:

- **Remove last plan** - removes the last path plan that was sent by the UAV to the mobile interface.
- **View camera** - switches into Camera View.



Figure 7: Pictures of the mobile user interface showing the Map View

- **View event log** - displays log messages sent by the UAV to the ground operator. Messages include feedback from the command or outcome of the mission.
- **Edit waypoints** - adds or removes labeled waypoints on the map (x,y,z,heading). Waypoints are stored in a database in persistent memory and can be used as parameters for the missions or commands e.g. fly to a waypoint or survey this building. All the waypoints saved in the database in previous sessions are loaded and displayed on the map at startup.
- **Add no-fly zones/Remove no-fly zones** - adds or removes areas that the UAV is not allowed to fly in. These areas are also taken into account during the path planning process. The user can choose an arbitrary convex polygonal area as a no-fly zone.
- **Mission start** - sends a command to start a preprogrammed mission.
- **Take off** - sends a command to take off.
- **Land** - sends a command to land with specified heading.
- **Yaw** - sends a command to change the heading of the UAV.
- **Climb** - sends a command to change the altitude of the helicopter.
- **Emergency brake** - sends a command to abort current flight mode, brake and switch into hovering mode.

The second and the third head menu options (labeled **Flyto** and **Command**, respectively) contain commands related to UAV control:

- **Fly to position** - sends a command to fly to a specified position.
- **Fly to waypoint** - sends a command to fly to a specified waypoint chosen from a database.
- **Fly to N positions** - sends a command to fly to a specified destination and a set of waypoints to pass through on the way.
- **Release camera lock** - sends a command deactivate a visual servoing camera mode.
- **Photogrammetry** - sends a command to start a photogrammetry mission. User can choose an arbitrary number of buildings from the current GIS database by clicking on the map. The goal of the mission is to fly to each of the building and take a picture of each facade.

All missions involving flying a trajectory, either directly e.g. by a “fly to” command, or indirectly e.g. during a landing procedure (approach to the landing position),

involve use of an onboard path planning service associated with the UAV. When a valid path is generated by this service, the UAV system sends its alternative to the mobile interface where it is displayed as an overlay on the current map in the display. All no-fly zones which were added using the mobile interface or which may have been added externally (e.g. by another ground operator or some deliberative service) are also taken into account during the path planning process. If a no-fly zone is added or removed in the UAV system database, this information is sent to the mobile interface, so that the ground operator always has an up to date view of the mission environment.

The last head menu option (labeled **Orientation**) contains options to rotate the display on the phone to 0, 90, 180 or 270 degrees.

Telemetry data and the last logged message are displayed at the bottom of the Map View. The telemetry data is sent by the UAV with a fixed rate and includes current sensor values such as position, heading and velocity of the UAV as well as the current flight mode. The UAV is displayed in the display as a green arrow indicating heading on the map.



Figure 8: Pictures of the mobile user interface showing the Camera View

In the Camera View, the main window is reserved for images downloaded from the UAV's camera (Figure 8). In the head menu option (labeled **Camera**) at the top, the user can request the current image from the camera

or start streaming video from the camera down to the mobile interface. At the bottom of the view telemetry data is displayed together with a virtual joystick. The joystick is used for control of the camera pan and tilt angles. It is also possible in this mode to rotate the display to 0, 90, 180, 270 degrees.

The mobile interface was tested on a Sony Ericsson P900 smartphone, which has GSM/GPRS class 8 support (4+1 slots). It also supports HSCSD transmission with 2+1 slots. The theoretical throughput for GPRS and HSCSD connections is presented in Table 2 and Table 1 respectively. In addition, the phone has a touch screen, a 16MB internal memory and is J2ME enabled. It uses the Connected Limited Device Configuration 1.0 (CLDC 1.0), which implies no floating-point support. All operations on floating points are done using fix-point integer calculations. On top of the CLDC 1.0 is the Mobile Interface Device Profile 2.0 (MIDP 2.0) (10) which is required for accessing any communication protocol lower than HTTP.

The communication protocol between the mobile interface and the UAV is based on the use of UDP datagrams. Its implementation is based on the results of tests from section 4. Package size varies depending on what kind of data is transmitted. For instance, images are sent in 512 bytes packages and commands using 16 or 64 bytes. Packages with commands are acknowledged and re-sent in case they are lost.

Additionally, a GPS receiver is connected via a bluetooth link, so the position of the ground operator can also be displayed on the map.

6 Flight test verification

To verify the potential for using GSM technology as a communication basis in our UAV network, we performed many instances of two types of flight tests.

In the first flight test type, a ground operator used the multi-modal interface we developed on the Sony Ericsson P900 smartphone to control the UAV and the camera in an interactive manner.

A mock surveillance mission was set up in our urban flight test area in southern Sweden. The mission task was to fly to a certain area and gather still images of a particular location in that area where a large car accident had taken place and where the police were unable to

travel due to traffic chaos along the way. The ground operator (GOP) used the P900 to setup and complete the mission in the following manner:

1. The GOP first commanded the RMAX UAV to take-off from homebase and hover at a particular altitude.
2. While this was being done, the GOP received information about surrounding mountain terrain and wind gusts due to an incoming storm in the area. To avoid flying in these areas, the GOP specified a number of no-fly zones in the P900 interface as shown in the first screen view in figure 10.
3. The GOP then marked a destination location on the P900 map view to fly-to which included desired velocity and altitude. A motion planner onboard the RMAX UAV was automatically called to generate a collision free path from the current location of the UAV to the destination location. This mission path was then sent from the UAV to the P900 and could be viewed by the GOP on the P900 screen as shown in the second screen view in figure 10. After viewing the proposed path, the GOP clicked on the confirm button and the execute command was sent to the UAV to execute the path. Note that it avoided the no-fly zones programmed into the interface.
4. Upon arriving at the destination, the GOP interactively moved the camera using the P900 virtual pan/tilt display in figure 8 to get the best photos possible of the scene of the accident.
5. After collecting a sufficient number of photos and downloading them to the P900, which would be transferred to the police by email.
6. The GOP called and asked for clearance to fly back to homebase. The GOP was given permission to fly back but was also notified that an additional region in the direction of home base was dangerous to fly through due to a great deal of smoke from an oil fire associated with the accident. The GOP entered an additional no-fly zone using the P900 interface as shown in the third screen view in figure 10.
7. The GOP then entered the homebase location into the P900 interface and issued a fly-to command. In a similar manner as before, a new collision free path was generated and confirmed and the UAV flew back to home base. The final leg of the mission is shown in the fourth screen in view in

figure 10.

8. The GOP then entered a land command via the P900 and the helicopter landed autonomously at homebase.

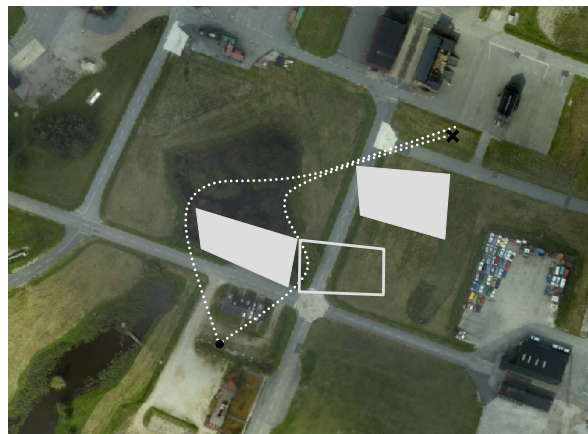


Figure 9: Summary of the first mission. Black cross marks the take off and landing position. Black dot marks the hovering position from which images were taken. Gray solid polygon area marks no-fly zones added before flying to the hovering position. Gray empty polygon area marks no-fly zones added before flying back to home base. White dotted lines mark flown trajectories.

Figure 9 shows a summary of the complete mission. It is important to observe the flexibility and ease with which this mission was planned and executed through the P900 using GPRS functionality and standard telephone functions such as phone calls and email. Note also, that theoretically, the ground operator could have been located almost anywhere and not necessarily near home base or even the vicinity of the accident.

In the second type of flight test, we tested the distribution of some of the RMAX UAVs onboard software. Recall that the basic architecture for the RMAX UAV is based on the use of CORBA and that much of the high-level deliberative functionality is structured as CORBA servers with a specific functional interface defined using IDL. Because of this, any functionality which is structured as a CORBA server, such as the motion planning functionality, could actually be placed anywhere on the Internet, or in the ground station, or on another UAV without any real effect in the operational mode of the UAV except perhaps for some additional latency in use of the functionality. For this flight test type, we placed the path planning service in different physical locations.



Figure 10: Pictures of the mobile interface during the flight test

After the UAV system startup, the path planning service started on another machine and CORBA register in the on-board hardware/software system. During the flight test when the path planner was needed, the UAV system used CORBA calls to invoke methods transparently from the planning service remotely on the other machine the same way as the service would have been used in the original DRC. Several different locations for the path planning service were used on the Internet and were tested successfully. We also ran a two UAV mission where the 1st UAV transparently called a motion planner on another UAV. In all cases our GPRS data link behaved adequately when using the path planning service remotely.

7 Conclusions

A GSM based communication network with a multi-modal interface implemented on a Sony Ericsson P900 mobile telephone was designed, constructed and implemented in an experimental system using two Yamaha RMAX helicopter platforms. Based on actual flight tests and empirical evaluation, we believe this type of technology and the use of mobile telephones as interfaces offers much promise for a wide variety of mission applications.

Based on our analysis and experimentation there are a number of existing benefits and several we can envisage

as the technology matures. The approach uses a mature, commercially available communication infrastructure which will continue to be developed, refined and supported in the future. It permits the operation of UAVs at large distances, out-of-sight from the ground operator. It provides a good redundant system alternative in the case where other communication frequencies jam. It provides a means of indoor communication when using micro air vehicles inside building structures. It also gives access to low price use of the WWW. This is an important point since there will be an increasing demand for using information on the Internet in real-time during mission execution.

We did notice a number of disadvantages with the technology, but in this case, we believe these disadvantages will be ameliorated in the near future. The bandwidth using GPRS is currently too narrow to permit real-time video streaming even with small resolutions, a functionality that we believe is quite important. Quite large delays also have to be taken into account during communication protocol design. The robustness and coverage of the GPRS technology outside of large cities is not yet at a level where one could *guarantee* safety and reliability at a level suitable for flying in and over populated areas. This being said, we believe this is a promising technology and will continue developing it with our UAV systems in the future.

8 Acknowledgements

This research has been supported in part by grants from LinkLab (www.linklab.se) and from the Wallenberg Foundation, Sweden.

References

- [1] Stefano Avallone, Antonio Pescapé, and Giorgio Ventre. Analysis and experimentation of Internet Traffic Generator. In *Proc. of the Int. Conf. on Next Generation Teletraffic and Wired/Wireless Advanced Networking*, 2004.
- [2] Peter Benko, Gabor Malicsko, and Andras Veres. A Large-scale, Passive Analysis of End-to-End TCP Performance Over Gprs.
- [3] R. Chakravorty, J. Cartwright, and I. Pratt. Practical Experience With TCP over GPRS, 2002.
- [4] R. Chakravorty and I. Pratt. Performance Issues with General Packet Radio Service, 2002.
- [5] P. Doherty, P. Haslum, F. Heintz, T. Merz, T. Persson, and B. Wingman. A Distributed Architecture for Autonomous Unmanned Aerial Vehicle Experimentation. In *Proc. of the Int. Symp. on Distributed Autonomous Robotic Systems*, pages 221–230, 2004.
- [6] Gunnar Heine and Holger Sagkob. *GPRS Gateway to Third Generation Mobile Networks*. Artech House, Inc, 2003.
- [7] T. Merz. Building a System for Autonomous Aerial Robotics Research. In *Proc. of the IFAC Symp. on Intelligent Autonomous Vehicles*, 2004.
- [8] Micro Edition Java 2 Platform. <http://java.sun.com/j2me/>.
- [9] David L. Mills. Internet Time Synchronization: The Network Time Protocol. In *Zhonghua Yang and T. Anthony Marsland (Eds.), Global States and Time in Distributed Systems*. IEEE Computer Society Press, 1994.
- [10] Mobile Information Device Prole (MIDP). <http://java.sun.com/products/midp/>.
- [11] E. Nettleton, S. Thrun, and H. Durrant-Whyte. Decentralised SLAM with Low-Bandwidth Communication for Teams of Airborne Vehicles. In *Proc. of the International Conference on Field and Service Robotics*, 2003.
- [12] Mischa Schwartz. *Mobile Wireless Communications*. Cambridge University Press, 2005.
- [13] SR Telecom. Point-to-Multipoint: Effective Rural Broadband Wireless Access A Comparison with Fixed Cellular Technologies. white paper.