The story behind the story of Apollo 13.

Class note

Eduardo Beira
School of Engineering, University of Minho (Portugal)

Building on the knowledge developed in the Mercury and Gemini space programs of the early 1960s, the Apollo program resulted in the transforming moon landing by Apollo 11 on July 20, 1969. The successful Apollo 12 mission followed in December, 1969. Among the many successes and inspirations of all the Apollo missions, the failure of Apollo 13 stands out, nevertheless, as a triumph of spirit, of insight, and of engineering.

Apollo 13 was launched on April 11, 1970. The next day, however, an explosion occurred that damaged the spacecraft so severely that its mission to land on the moon was aborted. The immediate challenge was to figure out how to use the crippled spacecraft to return the crew safely to earth, which required considerable improvisation by the crew and the support workers on earth..

Ken Cox was Technical Manager of the Primary Flight Control Systems for the Apollo program in the mid and late 1960s, and later Shuttle Technical Manager for Integrated Guidance, Navigation and Control. He is now Assistant to the Director, Engineering Directorate for NASA at the Lyndon B. Johnson Space Center, Houston. This is his story behind the story of Apollo 13.

I got a call at my home

When the explosion occurred on board Apollo 13 in April of 1970, I got a call at my home across from The Manned Spacecraft Center (now called the Johnson Space Center) at about 1:00 in the morning and I was told to get to Mission Control right away. I listened to the whole drama unfold, and I was part of the process of figuring out what to do to bring the spacecraft and the astronauts safely back to earth. I was part of that activity on the ground, listening as the life support system began to drain out of the Command Module, as we realized that we must use the Lunar Module as a life boat.

When I saw the movie Apollo 13, the emotions, my experience, how people felt... Apollo 13 was a very emotional film for me to watch. It depicted, in my judgment, with very great credibility the spirit of everyone trying to save that mission and bring the astronauts back.

To do this required us to use the Apollo spacecraft in a way that it was not designed to be used, and I'm going to tell part of the story of how this was done.

They were built like airplanes had always been built

Most of the fundamental decisions about how to design the Apollo spacecraft were made in the early 1960's. The Apollo system consisted of three spacecraft, the Command Module, the attached Service Module, and the Lunar Module. The command and service modules were built by North American (later Rockwell), while the lunar module was built by Grumman.

One of the main design issues was that if you launched a rocket toward the moon you would swing around the moon and come back toward the earth, but you wouldn't make it to the earth and get a good entry. So Apollo was designed to inject into lunar orbit by firing the big engine of the Service Module, a major Delta V propulsion burn, and then to later achieve a good earth reentry trajectory with another engine firing.

The original designs for the engine control systems called for analog controls, because the Mercury and Gemini systems prior to Apollo were analog. They were built like airplanes had always been built. Digital computers as we know them today, digital systems, had never been built for an aircraft or a spacecraft.

None had never flown with digital systems

A major decision made in approximately 1964 after the original designs were essentially completed, because someone realized, "Wait a minute, computers are coming and they give us a lot more capability." So NASA made the very fundamental decision to go to digital primary control systems for all three modules, with backup analog systems which would be built as planned by the spacecraft contractors.

The primary system, though, was the beginning of the new digital avionics, digital flight control systems for Apollo. There was no precedent since previous spacecraft - none had never flown with digital systems. Because of my prior background in digital systems, I was given the responsibility to develop the primary control systems for all of the Apollo vehicles.

Since the primary contractors did not have experience in digital control, MIT Instrumentation Lab (later the name was changed to Charles Stark Draper Lab) was given responsibility for the primary control systems software, computer, and digital computer implementation that was used for all three modules, and my role was to manage the project.

Right from the beginning I had one heck of an integration job, and I did it from scratch. There was no precedent. There were no reports saying "Here's the way we have built these systems in the past", and so it was a brand new, open ball game.

The only way to do it was to stick to the tried and true

I had gotten advice from Grumman and from North American, but they both wanted to use the digital capability, the new capability with exactly the same filters, exactly the same gains, exactly the same feedback loops as an analog system. They both advised me that the only way to do it was to stick to the tried and true, the analog design techniques, and just digitize it.

I did not want to do that. I realized that there were inherent properties of digital systems that were not available in analog systems. There were some good properties and some bad properties, but if you used the digital systems right you got some capabilities that are not at all equivalent, positive capabilities. So there was a big,

roaring debate which I was right in the middle of, on this whole question of the philosophy of how do you develop digital systems.
The great advantage of going digital was flexibility. You could make your control systems independent of the hardware sensors and effectors. You could change your control parameters after receiving actual flight test data. For example, we found out later that we didn't model all of the atmospheric properties correctly at the initial edge of the atmosphere. No airplanes had ever gotten up to that altitude, so we didn't have any real flight data. For all of the spacecraft, the mass properties were constantly changing, but once you built an analog control system you had no flexibility to make adjustments. You would find out when you got up in flight that the basic modeling was different than what you designed to.

With analog control systems, you would have had to physically remove the hardware and build new hardware. But because we had a digital system, we immediately changed the programming after the first Apollo flight and adjusted for extra firings and extra activity on the flight control system. In a digital system, it's a snap. You just go in there and adjust a few figures. So it was definitely the right thing to do because it gave Apollo mission flexibility with extremely low impact on the overall program.
I should mention that the computers that we were dealing with were very primitive by today's standards. The total memory of the onboard computers was about 32 K, and flight control only took about 35% - 40% of that. The rest of the code was concerned with navigation, guidance, targeting, and communication between earth and the spacecraft. It was programmed in HAL, which was developed at MIT.

What happens if you cannot use the big engine?

So now we are in late 1967, about 21 months before the first moon landing with Apollo 11, and 30 months before Apollo 13. We were working on the digital control system, and the people at MIT and I discussed the idea that we really ought to have a contingency mode for coming back from the moon if something happened and you could not fire the big command service module engine. What happens if there is a problem, and you cannot use the big engine? That was the drama of Apollo 13. Do we have contingency flight control, or contingency capabilities if something happened and we could not use the big engine? If you had the lunar module and command service module docked, the main engine on the lunar module could provide the incremental burn that was required so that we could get on the right trajectory to come back to earth.

We said, "Look, we think we have the time to add this capability to the digital control system, and we don't think it's that big a deal." It wasn't like we were running hell bent for leather and we were up against schedule constraints, and so I pushed real hard to say, "Well, let's put it in, damn it. This is an enhanced capability if something happens."

Now mind you, we could not define with any credibility or any predictability what the probability was that something would go wrong with the big engine, but it was obvious to us that if anything happened to the command service module and the big engine was not available to make this required burn in order to loop around the moon and come back, the spacecraft just should have the capability to use the engine in the lunar module.

I had even talked to some of my counterparts in the propulsion engineering design group, and they said, "Oh, no, we would never have an explosion like that. No, no, no. That's not a credible scenario."

Based purely upon good design practice and prior experience, there was no specific reason to protect against this happening. But we went ahead and we designed the thing anyway. We had the capability, and to us it was just the right thing to do.

What is the problem, and what is the probability?

Then we had to decide whether or not to include this contingency flight control capability on the lunar module into the backup analog flight control systems. We brought it up with Grumman, and they said, "In order for us to have the backup analog system, it's going to cause hardware to be changed and we cannot afford the impact on our schedule."

So we said, "O.K. we won't require it to be a function of the backup analog system, but the digital part can be done without changing hardware, or even changing code in the computer.

We took this idea to the Apollo program office, but they wanted us to prove that something might happen: "What is the problem, and what is the probability?" We didn't have the foggiest idea what the probability was that something might explode on the way over. So the initial response was, "Well, but you haven't proved yet that it is really needed."

Your request is disapproved

But I went ahead and requested the Lab to go ahead and code it for simulation and testing. When we had done the software coding and knew that it wasn't that big a deal, that it would work, I brought this issue up before the Apollo Software Control Board, which was run by Chris Craft (before he became head of the center).

I made an impassioned plea to put it in, and I really believed that it was important enough, and it was logical enough, and even though we didn't have the explicit criteria for what we were protecting against, I made the argument this it was the right thing to do.

Chris listened to all this at the formal software control board meeting, and much to my surprise and chagrin, he said, "Ken, I think that you've done good work here, but you haven't proven that you need it, and therefore your request to put this in the basic capability of the Apollo program is disapproved."

I couldn't believe it. I said to myself, "I don't give a damn whether I can prove it or not! It's the right thing to do!" I was just crushed. But as I was walking out the door to leave, Chris motioned me over to the other side of the room, and he got me in a corner where there was just me and him. He looked me right in the eye, and there was a twinkle in his eye, and he said "Put that mother in as soon as you can."

Immediately I realized that because of project politics, Chris did not want to open the gates for a lot of other changes that people had proposed that were not nearly as important as this.

Well, if that's the case ...

So I called the MIT Instrumentation Lab and I said, "Put it in, put it in!" At this point it was a joint thing, it wasn't just me directing them. This was an interesting relationship between a civil service person and a contractor out to do something. We were totally, absolutely committed in a partnership sense to doing the right thing.

We agreed that we would put the design in and test it in the main program. Now this was totally against the rules, totally against the bureaucratic trend, but we did it. And when I came back three or four months later to the Software Control Board, I said, "We have done this action, we have put it in the main line configuration control and if you turn this proposal down at this point, it will impact the program because you will have to take it out." And Chris had a twinkle in his eye, and he just said, "Well, if that's the case, I think we just ought to keep it in and accept the design." So that's how it was done.

We had done the right thing.

When the explosion occurred on Apollo 13 in April of 1970, it did render the main propulsion system of the command service module inoperable. Had we tried to use that engine it probably would have exploded. It was definitely not something we could have risked. So once they transferred to the lunar module, it became abundantly clear that we had done the right thing.

Of course I was very happy because I knew that I'd made a major contribution to the program itself. That's part of the intrinsic spirit and working together as a collective community that just flowed in the Apollo program. I think we were able to accomplish things like this because we had a very fluid organization in Apollo. We all had clearly defined goals, and we knew that this was an important, international and national endeavor.

We're second guessing history here, but I believe that it is probably the case, just in a probabilistic sense, that if we had not gone ahead and developed the digital control systems, Apollo 13 probably would not have made it back to earth.

Excerpts from Miller, W. and L. Morris, *"4th generation R&D. Managing knowledge, technology and innovation"*, John Wiley, 1999

MIT Portugal
Universidade do Minho

Class note prepared by Professor Eduardo Beira for “Innovation” unit, PhD Program, School of Engineering, University of Minho (2009)

V1, 16 November 2009