

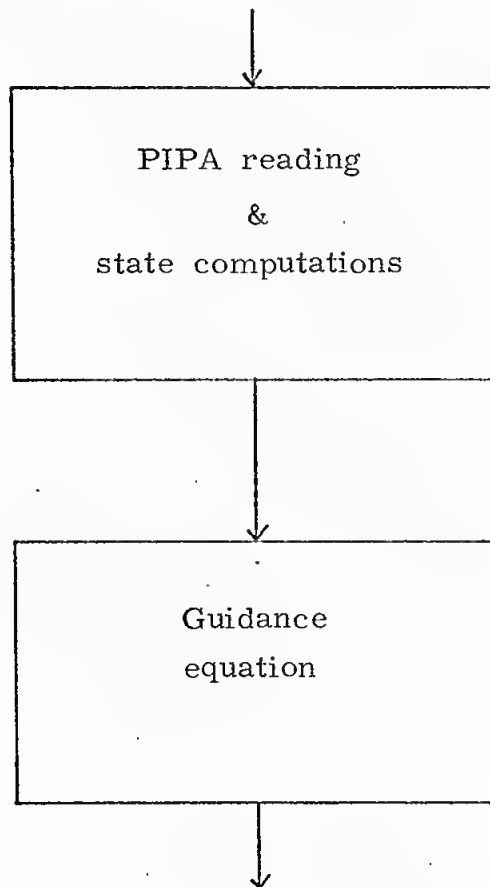
8-5-70

DEFINITION AND DESCRIPTION OF THE
VARIABLE GUIDANCE PERIOD SERVICER

Eyles

What is Servicer?

Servicer is a routine which runs periodically during powered flight in which navigation and guidance are performed. Servicer begins with the reading of the accelerometers, the average-G equation, and other vehicle state computations. This part is common to all the powered flight phases. Servicer next transfers control to the appropriate guidance equation and concludes after the output of displays and of commands to the autopilot and throttle.



Fixed and Variable Servicers

If the reading of the PIPAs occurs at a constant frequency the Servicer is said to be fixed. Guidancee period does not vary. If, on the other hand, the time of reading the PIPAs can vary in response to variations in the length of the Servicer computations, the Servicer is variable. In this case guidancee period changes and must be computed instead of being assumed.

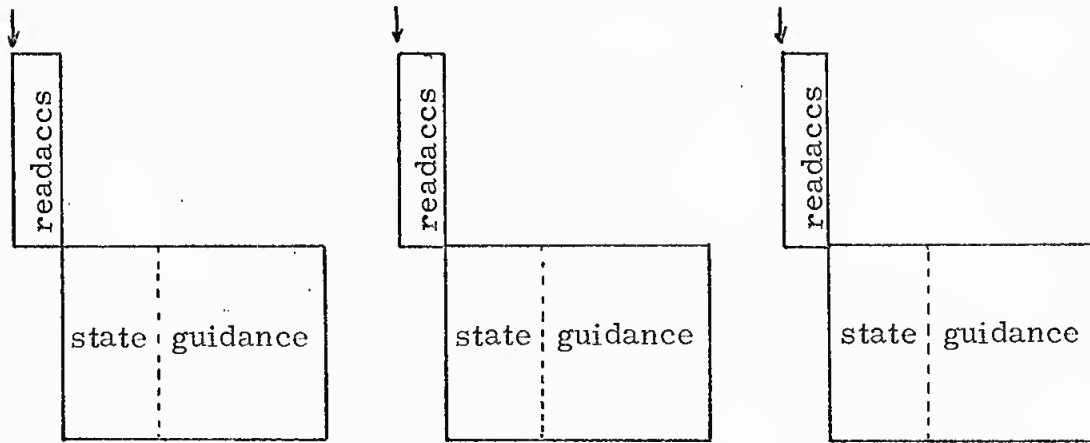
LUMINARY's Servicer

The Servicer now in LUMINARY is a fixed Servicer which runs with a period of 2 seconds. It is started every pass by an interrupt -- the READACCS task -- in which the PIPAs are read and a Servicer job set up. This job is set up whether or not the job initiated by the previous READACCS task has been completed. When TLOSS or extra computations cause the Servicer job to expand to the extent that it is not completed before the next READACCS, the next Servicer job begins anyway. The remainder of the first Servicer job is stored in the core sets and vae areas to be executed later, whenever there is time. This means (1) that the work at the end of the Servicer job does not get done, (2) that dangerous and very complex behavior may occur when odd left-over pieces of Servicer jobs at last get executed, and (3) that eventually software restarts result when core sets (alarm 31202) or vae areas (alarm 31201) are exhausted.

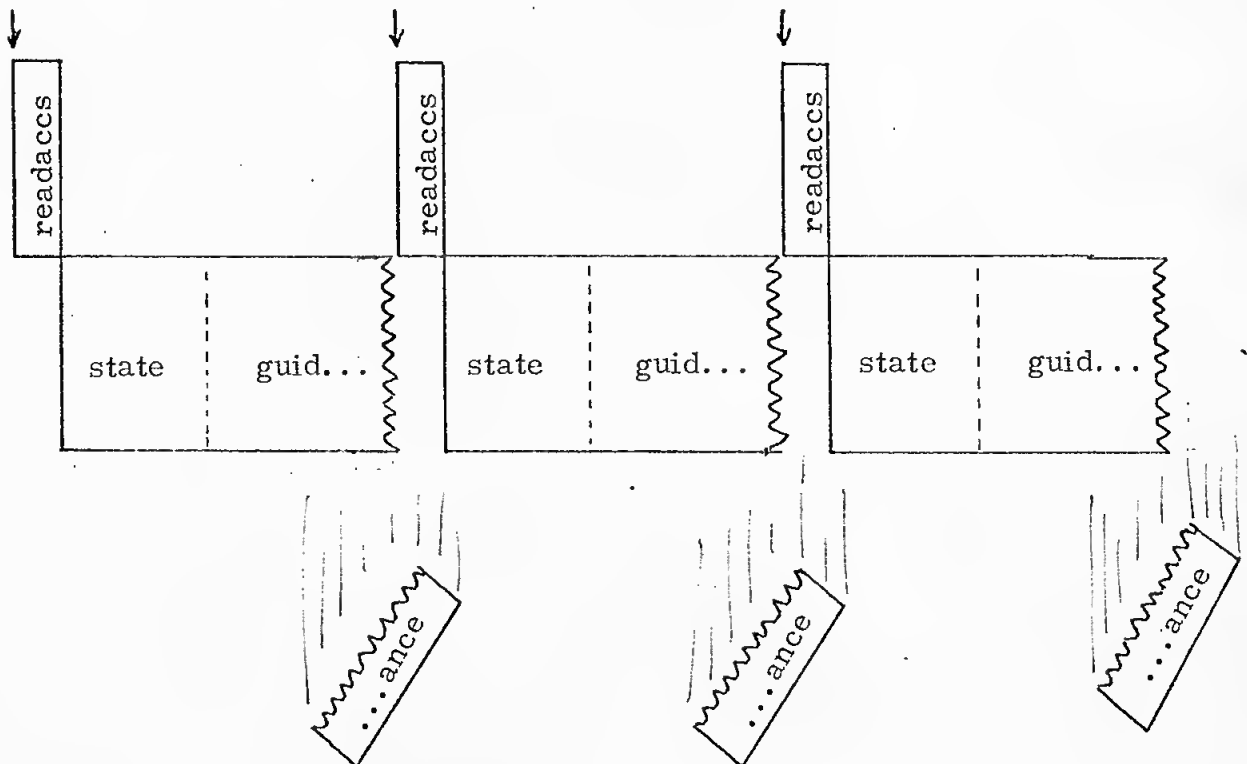
Here are pictures of the LUMINARY Servicer, in normal operation (A), and in trouble thanks to TLOSS (B).

TIME 0 1 2 3 4 5 6

A



B



ZERLINA History

The Servicer developed in ZERLINA is a second-generation variable Servicer. The first was developed over a year ago in another off-line version, DIANA by MOONLIGHT. Here guidance period was not continuously variable, but it could be changed by the astronaut in increments of 1/4 second by means of extended verbs. Despite its shortcomings this version contributes to the present effort (1) an idea of the difficulties and the magnitude of the effort involved in incorporating a variable Servicer, and (2) some specific blocks of coding, notably the average-G equations. Landings and the P40s were tested in this early version.

ZERLINA by ZOROASTER originated as a version of LUMINARY revision 145 in February of this year. By revision 18 it contained a variable Servicer essentially complete and tested in all the powered-flight phases of the nominal mission. In later revisions of ZERLINA a new Landing Analog Displays routine was developed which has already found its way into LUMINARY. Throughout, ZERLINA has been kept up to date with the main-line LUMINARY and ZERLINA revision 35 is the equivalent of LUMINARY revision 174 except for its superior Servicer. ZERLINA shares most of its subroutines with LUMINARY.

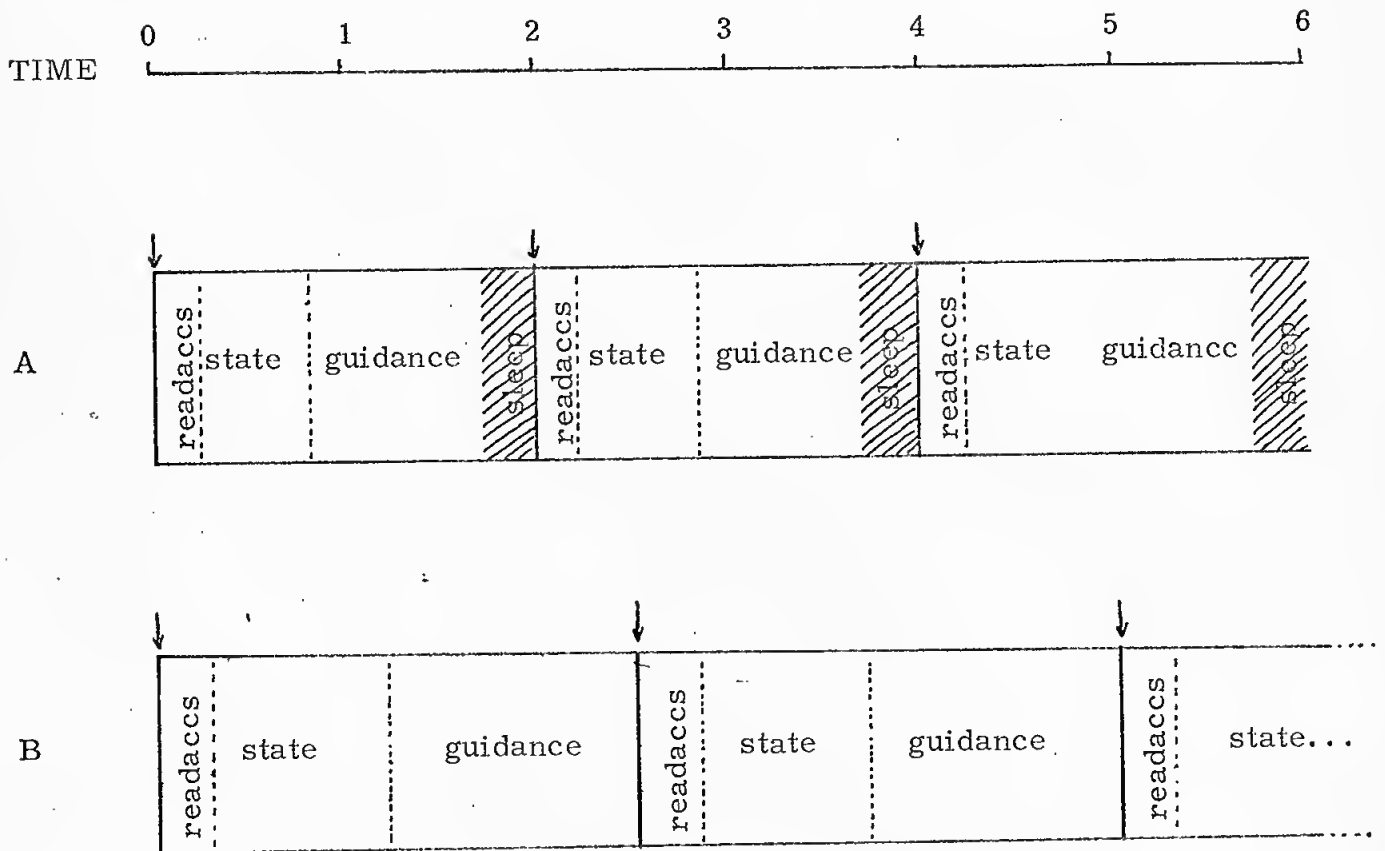
ZERLINA's Servicer

The variable Servicer developed in ZERLINA runs with a minimum period of 2 seconds. In normal operation when its computations are completed the Servicer job sleeps until the 2 seconds are up. But if TLOSS or extra work expand the Servicer duty cycle to the extent that it takes more than 2 seconds, the start of the succeeding Servicer cycle is slipped until it finishes. Guidance period is automatically stretched to fit the required duties and sleep time is eliminated. Because in this case the Servicer job is always active — at priority 20 — the other jobs that must be executed are given higher priorities to enable them to break in.

Since it can vary, guidance period is computed every pass. This in turn frees Servicer from rigid timing constraints and so the READACCS task can be eliminated. The PIPAs are read in the Servicer job. It makes restart protection casier and increases Servicer's ruggedness under time-stress.

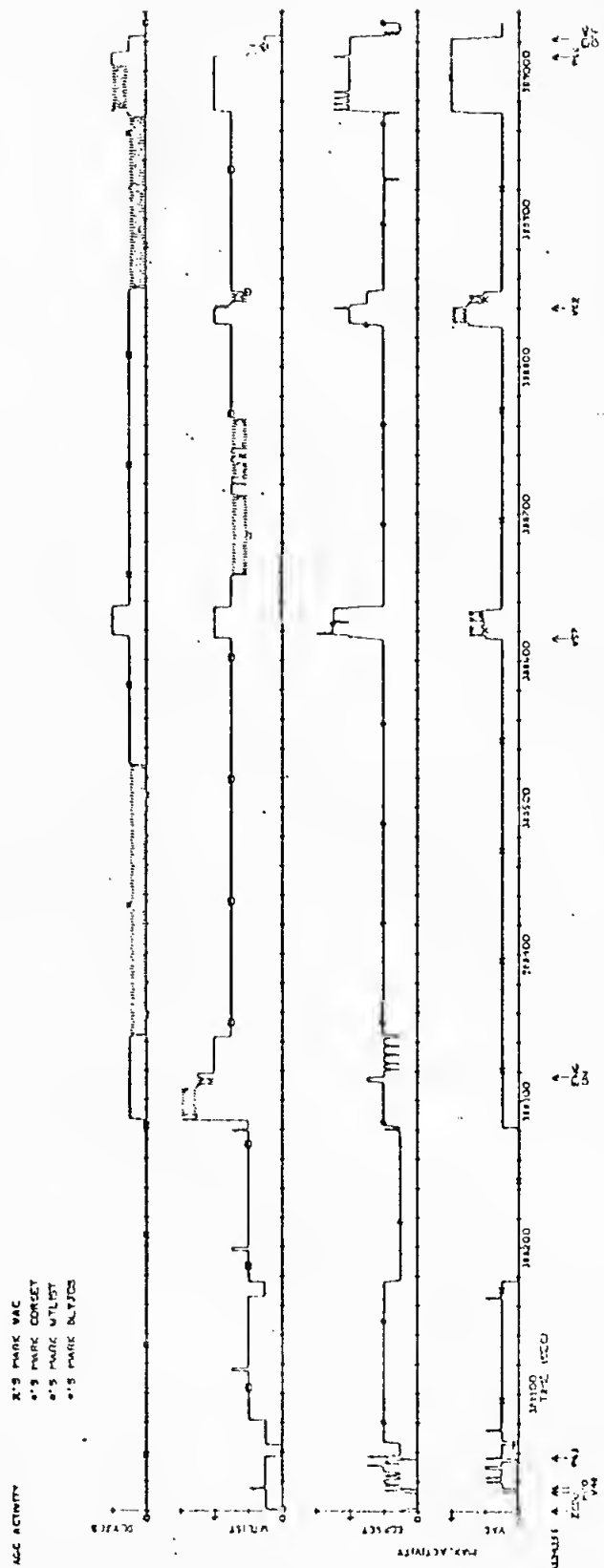
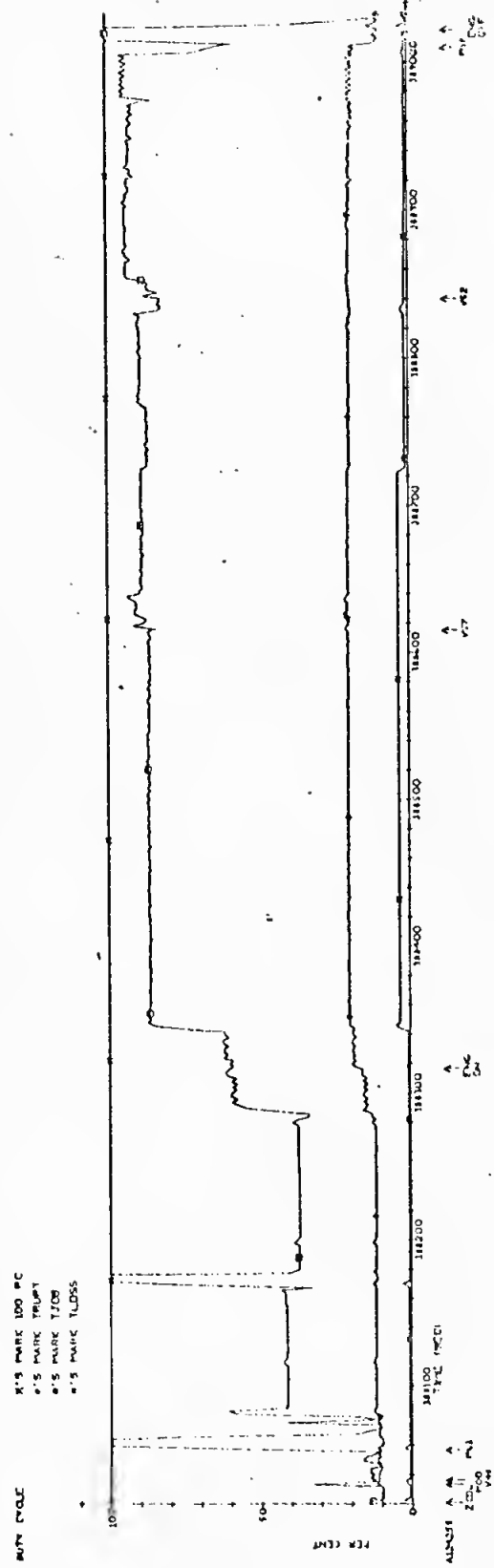
In ZERLINA the Servicer job runs end-to-end and thus cannot overlap itself. When it finishes it simply starts over. This means that the guidance equation — whichever is connected — must after issuing its commands transfer control to the beginning of Servicer, marked by the tag PIPCYLE, instead of ending the job.

Here are pictures of the ZERLINA Servicer, in normal operation (A), and adjusting to high TLOSS (B).



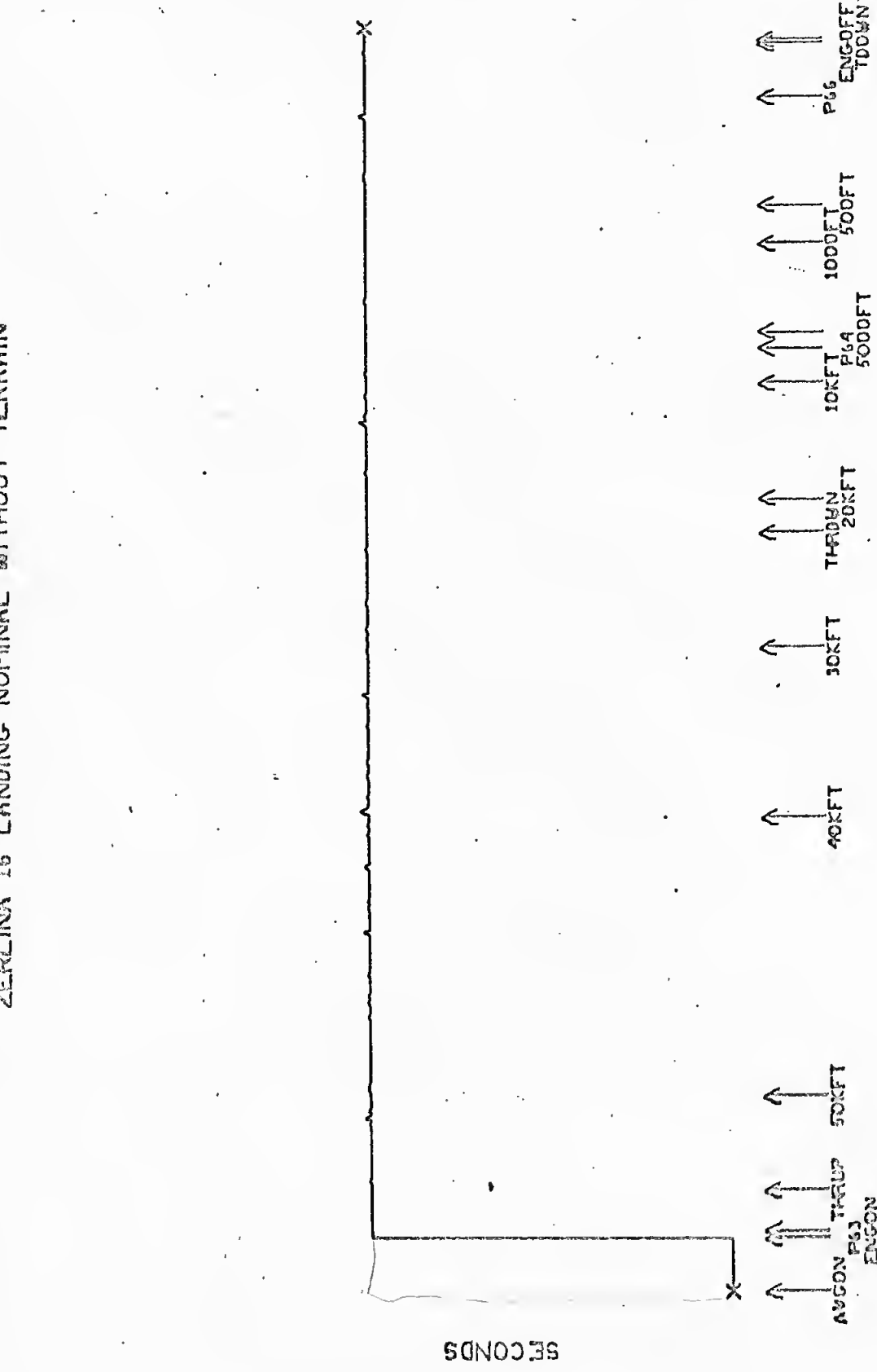
On the next pages are duty cycle and guidance period plots for Landings run on ZERLINA with zero and with 20% TLOSS. After that comes a flowchart of the Servicer job and its auxiliaries.

ZERLINA LANDING: NO TLCS



GUIDANCE PERIOD

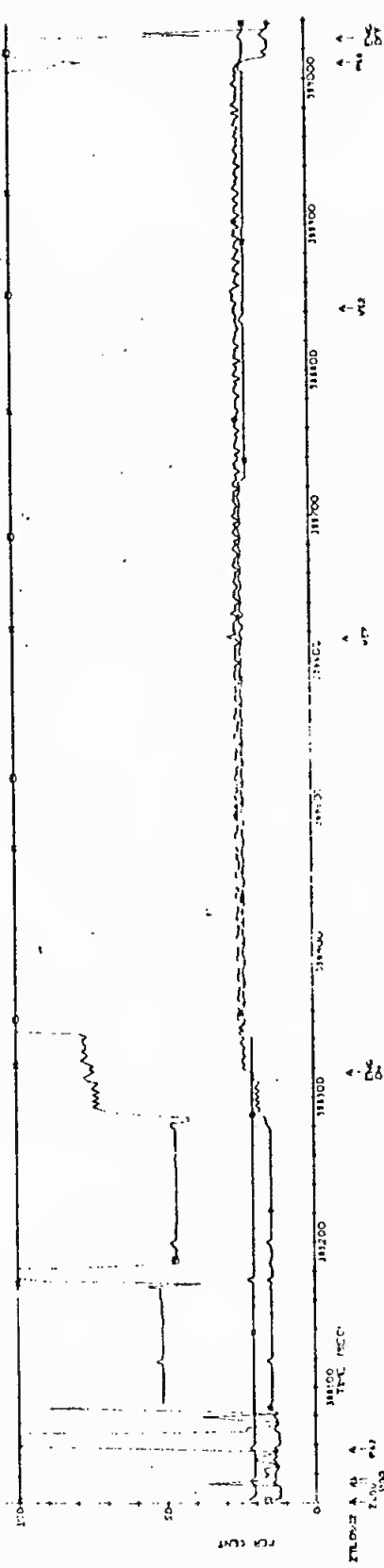
MARSROT NUMBER 08706095
 ZERLINA 16 LANDING NOMINAL WITHOUT TERRAIN



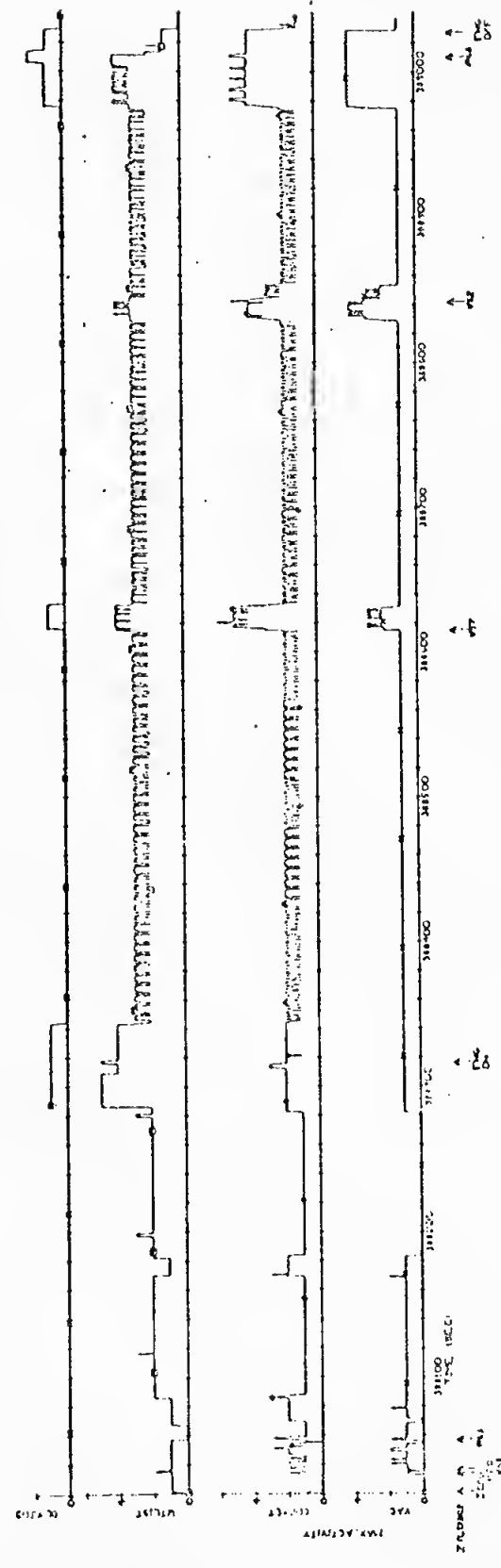
SECONDS G. E. T. 388300 388400 388500 388600 388700 388800 388900 389000

ZERLINA LANDING: 20% TLOSS

01'S MARK 100 PC
 02'S MARK 100 PC
 03'S MARK 100 PC
 04'S MARK 100 PC
 05'S MARK 100 PC



01'S MARK VAC
 02'S MARK COMBET
 03'S MARK UTILIST
 04'S MARK BLAZED



GUIDANCE PERIOD

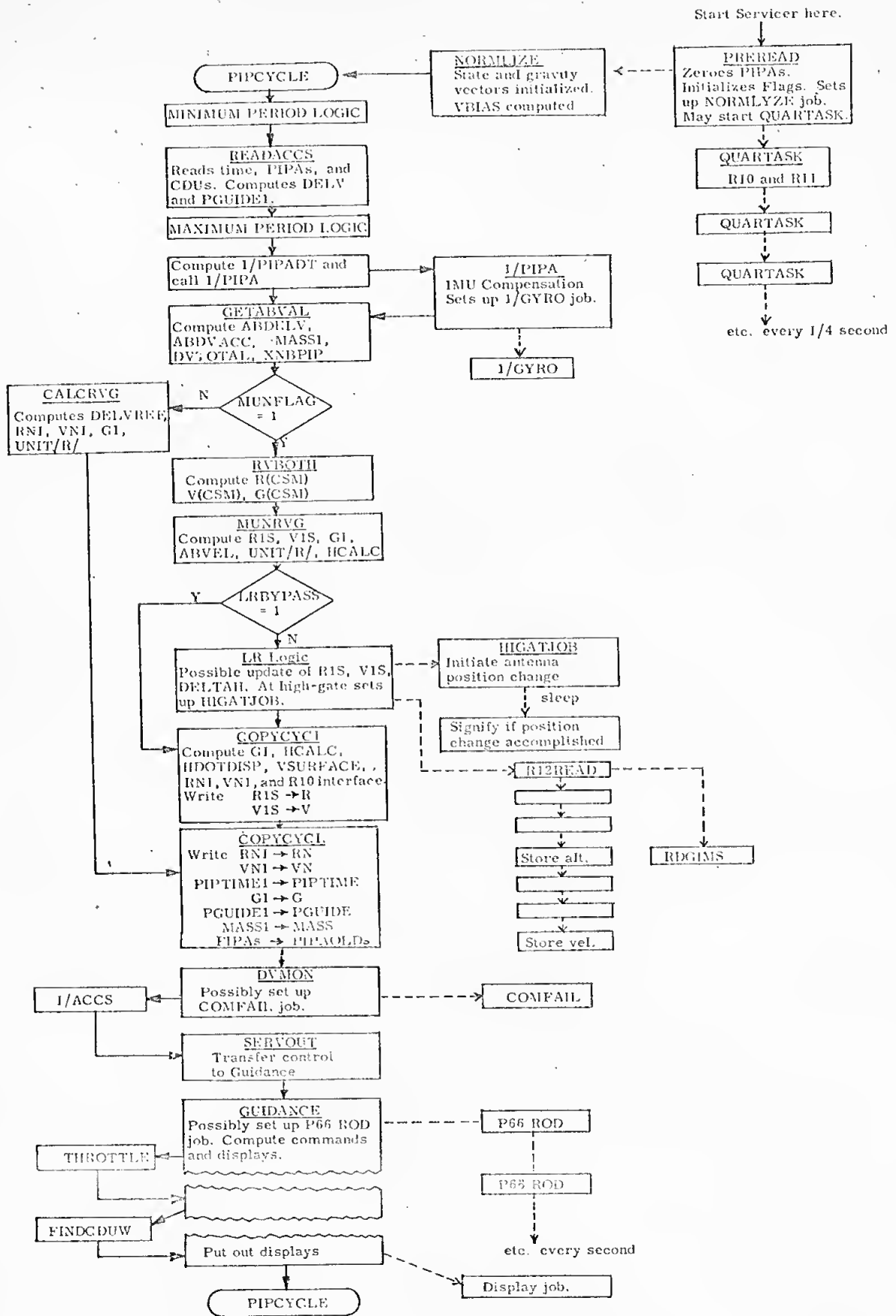
MARSROT NUMBER: 00720472
ZERLINA 16 LANDING WITH 20 PERCENT TLOSS

Handwritten notes:
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000



- ↑ AVCON P63 THRU UP 50KFT ENGCON
- ↑ 40KFT
- ↑ 30KFT
- ↑ THROWN 20KFT
- ↑ 10KFT P64 5000FT
- ↑ 10000FT 500FT
- ↑ P66 ENG OFF 1000N

SECONDS G. E. T. 388300 388400 388500 388600 388700 388800 388900 389000



Variable Servicer Topics

PREREAD is changed in one way. Formerly the task in which the Landing Analog Displays, the Abort Monitor, and certain Landing Radar functions are performed (called QUARTASK in ZERLINA) was set up for 8 passes by the READACCS task. Now it is set up for an indefinite number of passes by PREREAD.

Minimum Period Logic. First to be executed after P1PCYCLE is the Minimum Period Logic: if it has been less than PGMIN since the last PIPA reading then Servicer goes to sleep for the remaining time. PGMIN is 2 seconds. The reasons for such logic are (1) to establish a (hard) lower bound on guidance period for scaling purposes, (2) to provide a mechanism for granting (but not guaranteeing) some time to low priority extended verbs, and (3) to insure enough time for at least one full downlist to be sent for each Servicer cycle. Picking PGMIN to be 2 seconds has the additional advantage of making it unnecessary to remove the assumption of a 2 second period from the P40s guidance equations. The margin during these phases is so great that even the TLOSS at which the DAP breaks down is insufficient to make PGUIDE exceed 2 seconds.

Cyclical PIPA Reader is the name of one of the devices which made the writing of this variable Servicer somewhat easier. This works as follows: after PREREAD the PIPAs are never zeroed and each DELV is computed, not read from the PIPA, as the difference between this and the previous value of the PIPA, stored in PIPAOLD. Correction for the chance that the PIPA in question overflowed

between the two readings is easily performed under the comfortable assumption that 81.92 m/s cannot be honestly accumulated during one PIPA interval. (With DPS thrusting full-bore and both DPS and APS tanks empty it would take over 8 seconds to build up this velocity.) Advantages of this PIPA reading philosophy are (1) the straightforwardness it gives the PIPA reader restart protection, and (2) that it lets asynchronous routines like the P66 ROD equation work with a completely different PIPA interval without the interference of a periodic zeroing of the PIPAs by Servicer.

Maximum Period Logic. Here if PGUIDE1 exceeds PGMAX alarm 555 is issued. This is a plain alarm, not a POODOO or BAILOUT. This logic establishes a (soft) upper limit on guidance period for scaling purposes. 5 seconds is a reasonable value for PGMAX.

1/PIPA. Guidance period is input to the IMU compensation routine 1/PIPA (and 1/GYRO) through the register 1/PIPADT, which is computed every pass from PGUIDE1. The scaling of 1/PIPADT had to be changed from units of 2^8 centiseconds to units of 2^{10} .

ABDVACC. ABDELV has always contained delta-V magnitude over the PIPA interval. With the PIPA interval a constant 2 seconds ABDELV could be used as an acceleration, but with guidance period varying it cannot be. So it is supplemented by ABDVACC, a true acceleration computed from ABDELV by multiplying it by $2\text{SECS}/\text{PGUIDE1}$ for use by the delta-V Monitor, the Ascent Guidance (in place of ABDVCONV), and 1/ACCS.

Average-G. The average-G equations were modified to take account of guidance period and to compute \underline{G} , a gravitational acceleration in units of 2^{-6} m/cs/cs, instead of $\underline{GDT}/2$, a velocity in which the DT is an implicit 2 seconds. \underline{G} suits most users better than $\underline{GDT}/2$. The average-G equations in LUMINARY are:

$$\text{PGUIDE} = 2 \text{ seconds}$$

$$\underline{R1S} = \text{PGUIDE} (\underline{V} + \underline{\text{DELV}}/2 + \underline{\text{GDT}}/2) + \underline{R}$$

Compute $\underline{\text{GDT}}/2$ from $\underline{R1S}$.

$$\underline{V1S} = \underline{\text{GDT}}/2 + \underline{\text{DELV}}/2 + \underline{\text{GDT}}/2 + \underline{\text{DELV}}/2 + \underline{V}$$

In ZERLINA these become:

$$\text{PGUIDE1} = \text{PIPTIME1} - \text{PIPTIME}$$

$$\underline{R1S} = (\text{PGUIDE1} \underline{G}/2 + \underline{\text{DELV}}/2 + \underline{V}) \text{PGUIDE1} + \underline{R}$$

Compute $\underline{G1}$ from $\underline{R1S}$.

$$\underline{V1S} = \text{PGUIDE1} \underline{G1}/2 + \text{PGUIDE1} \underline{G}/2 + \underline{\text{DELV}}/2 + \underline{\text{DELV}}/2 + \underline{V}$$

Landing Radar Incorporation Logic. Changes in the landing radar portion of Servicer include the addition of a state vector integration to find LM altitude and velocity at the time of the radar read. This is necessary because R12READ cannot be synchronized with PIPTIME, as it is in current LUMINARYs, because PIPTIME is not known in advance. Also a Nav Base to Stable Member transformation must be performed on the antenna beam vectors which is valid at the time of the radar read.

RDGIMS, a task set up by R12READ, records the PIPAs, the CDUs, and time (in register LRTIME) at the mid-point of the read. Using consistent values for the time of the read Servicer computes a position and a velocity correction which are used to modify R1S and V1S. These changes are almost like putting back coding which existed in LUMINARY 1C and earlier when the velocity read was not synchronized. This is the most time consuming of the extra computations required by the variable Servicer, but it does not prevent Landings from running at the minimum period of 2 seconds throughout in the absence of TLOSS. During Landing Radar operation the CDUs and time for the radar read sent out on the downlist are the values recorded in RDGIMS, not the values at PIPTIME as in LUMINARY. Finally, HCALC1 is used for altitude at the time of the radar read. The register HCALC is used for DSKY display, in P66 as in the other phases. In LUMINARY, for no good reason, both are used for display.

Changes outside Servicer

Restart Tables. Servicer in ZERLINA has only a job to restart protect, the READACCS task having been eliminated, and so the special variable phase-change routine SERVCHNG and the type of PHASCHNG in which the restart point is specified by a 2CADR in-line are all that is needed in establishing restart points. Thus all the group 5 restart tables can be deleted.

Pinball Noun Tables. Because HCALC1 is only altitude at the time of the radar read and HCALC is at all times the altitude suitable for display on the DSKY, nouns 60, 63 and 92 should have HCALC substituted for HCALC1.

Landing Guidance Equations. The Landing guidance has to be modified to use G instead of $GDT/2$. Additionally, in the same section, priority is raised to 23 before the display routine is called and lowered again to 20 afterwards. This is so the off-line display job will have sufficient priority to break in on the Servicer job and post its displays even in the presense of TLOSS. Also, the VACRLEAS calls at the end of the Landing guidance are removed. Otherwise Servicer would start again at PIPCYCLE without a vac area.

P66 Guidance. Of the two P66 ROD computations performed every 2 seconds, in LUMINARY one is part of the Servicer job and the other is separate. In ZERLINA both are outside of the Servicer job in an asynchronous loop with a 1 second period. This loop is initiated and initialized when P66 is selected. The P66 ROD equation is executed every 1 second regardless of how slowly Servicer is running. The P66 horizontal control equation remains part of the Servicer job and will be executed less often when there is high TLOSS. In the P66 ROD equation \underline{G} takes the place of $\underline{GDT}/2$ and adjustments are made to adapt it to the Cyclical PIPA Reader.

Throttle Control Routine. Because a 2 second guidance period is implicit both in ABDELV and in FWEIGHT (a thrust compensation number, but really proportional to velocity not acceleration), both used in the computation of present thrust, this computation is changed. In LUMINARY it is

$$FP = ABDELV \text{ MASS} + FWEIGHT.$$

In ZERLINA it becomes

$$FP = (ABDELV \text{ MASS} + FWEIGHT) 2SECS/PGUIDE.$$

FINDCDUW wishes to complete the commanded attitude manoeuvre in exactly one guidance period. Thus, if guidance period expands, the rate commanded by FINDCDUW must be diminished. This is done by the usual method of multiplying by 2SECS/PGUIDE. Thus the guidance period from the preceding full pass is used to predict the period for the pass about to begin.

When guidance period is fluctuating from pass to pass, as it sometimes does in P66, this prediction may be wrong and an attitude overshoot may result. However the time constant of the P66 horizontal equation (5 seconds) is sufficiently long to assure stability.

Ascent Guidance. In the Ascent Guidance routine G replaces GDT/2, ABDVACC replaces ABDVCONV as the input to the thrust magnitude filter, and, as in the Landing guidance, priority is raised before the display routine is called and lowered afterwards.

Landing Analog Displays are adapted to the Cyclical PIPA Reader. Also, since QUARTASK continues indefinitely once set up by PREREAD, there is no need to count passes in this routine.

Abort Interface. When a P70 or P71 abort is commanded a software restart is performed to flush the Landing guidance. In LUMINARY this guidance is protected in a different group than Servicer (group 3) and so the flushing is accomplished by turning off group 3 before the restart. In ZERLINA both Servicer and the guidance equation are protected in group 5. When Servicer transfers control to the guidance equation it sets the new SERVOVER flag; when control returns to Servicer SERVOVER is reset. Accordingly, before its software restart, the abort lead-in checks the SERVOVER flag and if it is set, indicating guidance in progress, the address of PIPCYCLE is substituted for the guidance address in the group 5 restart registers. Thus only Servicer, and not the guidance equation, is restarted after the software restart.

IMU Compensation. In 1/PIPA and 1/GYRO, the IMU compensation routines called by Servicer every pass, slight changes must be made to adapt to the new scaling of 1/PIPADT.

1/ACCS in LUMINARY uses ABDELV as an acceleration. In ZERLINA it is modified to use ABDVACC, a true acceleration, instead.

ZERLINA in Operation

The operational differences between ZERLINA and LUMINARY are in what will not happen, not in what will. The possibility of 31201 and 31202 alarms and anomalous behavior due to TLOSS is drastically diminished. V16 monitors can be used without fear in ZERLINA.

Alarm 1466, issued by LUMINARY when a P66 guidanee pass is dropped, does not exist in ZERLINA. P66 guidanee is never dropped. One new alarm docs exist in ZERLINA, already mentioned in the Maximum Period Logie: alarm 555. With PGMAX set at 5 seeonds this alarm will not occur with a TLOSS lower than 20%. If it did oeeur it would probably do so in P66 when guidanee period is greatest. In this ease rapid nulling of hõrizontal veloecity could not be expected from the P66 horizontal equation and probably attitude should be controlled manually in attitude-hold. ROD performanee will be unaffected because the P66 ROD equation continues to run every seeond.

Another alarm sometimes seen in very high TLOSS ZERLINA runs is alarm 32000. This alarm, with a software restart, is issued when the DAP eyele overlaps itself. It would appear in LUMINARY runs too if other things did not happen first. Testing so far indieates that the eorreet response to this alarm is to press on.

But I should end up by emphasizing that TLOSS will eause no alarm in ZERLINA until it reaches the neighborhood of 20%.