Magneplane International • Massachusetts Institute of Technology
United Engineers and Constructors • Raytheon Equipment Division
Failure Analysis Associates • Bromwell & Carrier
Beech Aircraft Corporation • Process Systems International

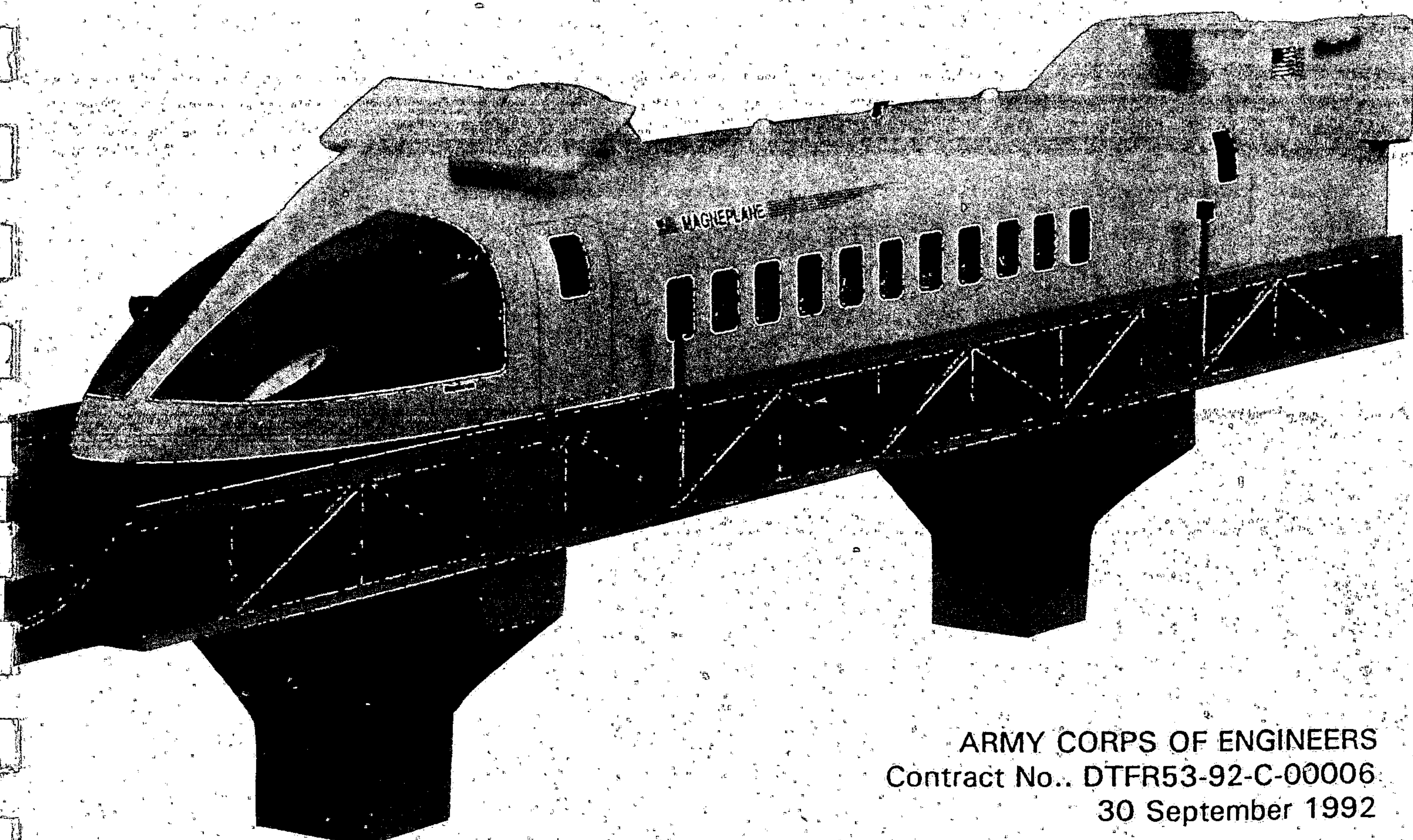
SYSTEM CONCEPT DEFINITION REPORT

for the

NATIONAL MAGLEV INITIATIVE

Volume 8

SUPPLEMENT D: VEHICLE SPECIFICATION
SUPPLEMENT E: LSM WINDING INDUCTANCE CALCULATION
SUPPLEMENT F: VEHICLE DYNAMIC RESPONSE EQUATIONS
SUPPLEMENT G: ROUTE ANALYSIS TOOLS
SUPPLEMENT H: HEAVE DAMPING CAPABILITY ANALYSIS
SUPPLEMENT I: BACKUP MATERIAL FOR CONTROL & COMMUNICATION



ARMY CORPS OF ENGINEERS
Contract No.. DTFR53-92-C-00006
30 September 1992

Magneplane International, Inc.
Jet Aviation Terminal, Hanscom Field West
Bedford, Massachusetts 01730
phone: 617 274 8750; fax: 617 274 8747

INDEX TO VOLUMES

LIMITED RIGHTS NOTICE

SUPPLEMENT D: VEHICLE SPECIFICATION

CONTENTS

SUPPLEMENT D
SECTION A

MODEL SPECIFICATION - MAGNEPLANE

MAGLEV SYSTEM CONCEPT DEFINITION

SECTION A

LIST OF APPLICABLE SPECIFICATIONS:

1) FAR (14CFR), PART 21

2) FAR (14CFR), PART 25

3) MIL-E-6051 D

4) FAA ADVISORY CIRCULAR 20-136

5) MIL-B-5087

6) MIL-F-1797

7) MIL-STD-1568A

8) FAA ADVISORY CIRCULAR 2053(b)

1.0 SCOPE (TITLE)

1.1 Scope. This document describes the proposed design configuration of the Magneplane high speed ground transportation vehicle as developed under U. S. Government Contract DTFR 53-92-C-00006 as a part of the National Maglev Initiative. The basic Magneplane vehicle is a 140-passenger vehicle capable of 300 mph speeds while operating on an elevated guideway. A smaller 45-passenger vehicle is also defined.

The Magneplane is magnetically levitated to operate above an elevated guideway and is propelled by means of a synchronous linear electrical motor concept in which super-conductive magnets in the vehicle interact with electrically-powered coils in the guideway to propel the vehicle at high speed.

Model Designation Magneplane - 140

Number of Crew - 1
Number of Passengers - 140
Propulsion - Linear
Synchronous Electric

Model Designation Magneplane - 45

Number of Crew - 1
Number of Passengers - 45
Propulsion - Linear
Synchronous Electric

1.2 Mission. The Magneplane vehicle is intended to provide high speed ground transportation over a network of elevated guideways which would link population centers. The Magneplane system is intended to meet transportation needs for route lengths of 400 sm (640 km) or less in order to relieve dependence on automobiles and airplanes for these route lengths. The Magneplane system is intended to move up to 25,000 passengers per hour over a single guideway and to operate at speeds exceeding 300 mph (134 m/s).

2.0 APPLICABLE DOCUMENTS (TITLE)

2.1 Effectivity of Documents. The documents referenced in this specification are only applicable to the extent specified. The documents specified herein shall be of the issue at the time this specification was written unless otherwise noted.

3.0 REQUIREMENTS (TITLE)

3.1 Vehicle Characteristics. The description of the vehicle characteristics consist of drawings, approval basis, systems and interfaces, performance, weights, center of gravity, areas, and other general dimensional data as listed in the following subparagraphs.

3.1.1 Drawings.

3.1.1.1 Two-View Drawing. See Figure A-1.

3.1.1.2 General Interior Arrangement. See Figures A-2, A-3 and A-4.

3.1.2 Vehicle Approval Basis. The Magneplane shall be certified for the commercial carriage of passengers to a set of regulations developed by the Federal Railroad Administration for high speed vehicles of this type. It is assumed these regulations will be similar to airline transport regulations (FAR Part 25) where applicable. Pending establishment of such regulations, some portions of FAR Part 25 have been used as an appropriate design standard.

This vehicle will be approved for carriage of passengers for hire in interstate commerce.

3.1.3 System Interfaces. The Magneplane shall be designed to be functionally and physically compatible with all Contractor Furnished Equipment (CFE) and any required Government Furnished Equipment (GFE) to the extent specified in the final detail specification.

3.1.4.1 Tabulated Performance.

	140	45
Max. Payload (Passengers & Baggage)	30,690 Lbs	10,080 Lbs
Max. Payload (Freight)	8,000 Lbs	---
Max. Speed (150 Meter/Sec)	335 mph	335 mph
Speed on 3-1/2% Grade (134 Meter/Sec.)	300 mph	300 mph

	140	45
Max. Payload (Passengers & Baggage)	13950 kg	4580 kg
Max. Payload (Freight)	3640 kg	---
Max. Speed (150 Meter/Sec)	150 m/s	150 m/s
Speed on 3-1/2% Grade	134 m/s	134 m/s

3.1.4.2 Payloads. The vehicle shall be capable of carrying simultaneously, the following payloads:

3.1.4.2.1 Passengers and Baggage. A total of 140 or 45 passengers with accompanying baggage. Design weight for this is 30,690 lbs (13950 kg) (140 passenger) and 10,080 lbs (4582 kg) (45 passenger).

3.1.4.2.2 Cargo Capability. Provisions for the carriage of up to 8,000 lbs

(3636 kg) of cargo shall be provided in the rear of the vehicle (140 passenger). These shall be limited to small parcels and high priority mail.

3.1.5 Weights.

3.1.5.1 Gross Weight. The Design Gross Weight is 105,000 lbs (47727 kg) (140 passenger) and 55,000 lbs (25000 kg) (45 passenger).

3.1.5.2 Empty Weight. The guaranteed empty weight shall be specified in the detail specification. The Design Empty Weight is 66,310 lbs (30140 kg) (140 passenger) and 44,920 lbs (20418 kg) (45 passenger).

3.1.6 Center of Gravity Locations. The design C.G. is from FS 56.0 to FS 70.0 (140 passenger), FS 34 to FS 40 (45 passenger).

3.1.7 Areas. See Figure A-5.

3.1.8 Dimensions and General Data. See Figures A-1, A-2, A-3 and A-4.

3.1.9 Control Surface and Related Control Movements. Control surface travel limits will be based on aerodynamic controllability requirements, hinge moments and actuator limits.

3.2 GENERAL DESIGN REQUIREMENTS

3.2.1 Vehicle Design and Construction. The basic Magneplane shall be designed and constructed to conform with applicable FRA and FAR (14CFR) Part 25 (or equivalent) requirements using best commercial practices.

The basic design criteria shall emphasize simplicity, ease of maintenance, reliability and safety consistent with operational effectiveness. In each case where conflict between operational requirements and basic design criteria exists, alternatives shall be carefully formulated and reviewed for an acceptable compromise.

3.2.1.1 **Selection of Materials and Standard Parts.** The materials, parts and processes for the vehicle shall be as required by MIL-HDBK 5 and FAR (14 CFR) Part 21 Subpart K, unless otherwise specified. Any variations shall be listed as exceptions to the basic certification. The use of composite material technology shall be emphasized in those applications where the characteristics are clearly superior.

3.2.2 **Environmental Conditions.** The Magneplane shall be designed to be capable of performing under the extremes of icing, heavy rainfall, saltwater-laden breezes, humidity, snow, ice, temperature, sand, and other environmental factors including the following:

- The minimum temperature for operation shall be -40 degrees Centigrade.

- The maximum temperature for operation shall be 40 degrees Centigrade.

3.2.3 **Survivability.** There are no specific survivability design requirements.

3.2.4 **Shipboard Compatibility.** Not applicable.

3.2.5 **Electromagnetic Compatibility (EMC).** The following are requirements for electromagnetic compatibility for the Magneplane.

3.2.5.1 **System EMC Requirements.** The vehicle, including all associated sub-systems and equipment shall be designed to achieve compatibility between all individual sub-systems and equipment in accordance with MIL-E-6051D and FAR (14CFR) Part 25. EMC shall be achieved between all individual sub-systems and equipment while operating simultaneously.

3.2.5.2 **Vehicle Grounding.** Grounding jacks are required for grounding the vehicle during maintenance operations.

3.2.5.3 **Electrostatic Discharge and Lightning Protection.** The vehicle and its systems shall be designed to the requirements of MIL-STD-461 to withstand the effects of static electricity and lightning without mission degradation or catastrophic failures. The requirements to meet the FAA HERF (High Energy Radiated Field) threat shall also be addressed. Design for EMP (Electromagnetic Pulse) is not part of the Preliminary Design configuration.

3.2.5.3 Electrostatic Discharge and Lightning Protection. (Continued).

Applicable Magneplane systems shall be designed to meet the lightning strike requirements of FAR (14CFR) Part 25 using FAA Advisory Circular 20-136 as a design guide. Electronic and electrical systems shall utilize current technology circuit design, shielding, wire routing, groundplanes, bonding and external/internal terminal protection devices as required to provide immunity from damage for all equipment providing mission critical/essential functions.

3.2.5.4 Electrical Bonding. Applicable systems and related structure, both metallic and composite, shall be designed to meet the requirements of MIL-B-5087.

3.3 AERODYNAMICS

3.3.1 General. The vehicle ride qualities shall not exceed the following limits:

<u>**Longitudinal Acceleration**</u>	
Maximum Normal Operation	0.16g
Maximum Emergency	0.6g
<u>**Lateral Acceleration**</u>	
Maximum Normal Operation	0.1g
<u>**Vertical Acceleration**</u>	
Maximum Normal Operation	Positive +1.2 (Vehicle Accelerated Up)
Maximum Normal Operation	Negative -0.2 (Vehicle Accelerated Down)

3.3.2 Stability and Control. Stability and Control characteristics shall meet the requirements of 3.3.1.

3.4 STRUCTURAL DESIGN CRITERIA

3.4.1 Loads. Vehicle design loads are based on design maximum vertical accelerations of 2.5g upward and 1.25 downward. An additional stiffness requirement is imposed that static deflection at rest cannot exceed one (1) inch (.025 m) at any point. Additional emergency requirements are imposed for seat attachment and floor structure. A 1.5 safety factor shall be applied to the design loads to determine the ultimate loads. The design loads are summarized below.

STATIC DESIGN LOADS
(Normal Structural Load Factor)

Vertical Up Acceleration	(2.5)
Vertical Down Acceleration	(1.25)
Lateral Acceleration	(1.5)
Longitudinal Acceleration	(1.5)

Emergency Structural Loads (Applied at seats)
(Load Factor)

Vertical Up Acceleration	3.0	(Applied to seat, vehicle restrained).	
Vertical Down Acceleration	6.0	"	"
Rearward Acceleration	1.5	"	"
Forward Acceleration	9.0	"	"
Lateral Accel.(Airframe)	3.0	"	"
Lateral Accel.(Seats & Attach)	4.0	"	"

Note: The normal loads shall be assumed to be encountered in service. The emergency loads are assumed to be encountered only in an emergency and need not be addressed from a repeated load standpoint. The emergency loads are applicable for design of occupant protection features. Structural damage may be sustained when the emergency loads are applied as long as the failures do not jeopardize the occupants.

3.4.2 Design and Construction. Design and construction shall meet all applicable D.O.T. regulations, including Federal Railroad Administration criteria and those portions of the Airline Transportation regulations deemed applicable.

3.4.3 Material, Processes and Parts. The materials, components, parts, and processes shall be required by FAR (14CFR) Part 21, Subpart K. The anti-ice/deice system shall not preclude normal painting of the exterior.

3.4.3.1 Corrosion Prevention and Control. Corrosion prevention and control measures shall meet the intent of MIL-STD-1568A.

3.4.4 Damage Tolerance and Service Life. The structure shall be designed for a service life of 25 years or 60,000 hours. All composite components utilized shall be capable to withstand impact threats normally encountered in service without visible damage.

3.4.5 Lightning Strike Criteria - Airframe. Airframe components shall be protected against catastrophic effects from lightning strike per FAR 25.581 and Advisory Circular 2053(b).

3.5 FUSELAGE (SUBTITLE)

3.5.1 Description. The fuselage consists of the main body of the vehicle including the crew station, cargo compartment, equipment compartments, landing gear bays, passenger and cargo doors and aft tailcone.

The fuselage structure is primarily a shell structure, fabricated of a graphite epoxy/nomex core, honeycomb sandwich.

3.5.2 Crew Stations. The Crew Station of the Magneplane provides for two (2) personnel to include the operator and observer. (Normal crew is one).

3.5.2.1 Windshield. There is no forward facing windshield.

3.5.3 Cargo Compartment. The cargo compartment consists of the portion of the fuselage extending approximately 10 ft (3 m) aft from the aft entry door. This compartment is shown in Figure A-1.

3.5.4 Doors and Hatches. All doors and hatches shall include safeguards against single element structural failures while in operation. The following doors and hatches are a part of the Magneplane design.

3.5.4.1 Crew Entry Doors. The crew shall enter through the normal passenger entry doors.

3.5.4.2 Cargo Door. Cargo and baggage are loaded through the aft passenger entry doors.

3.5.4.3 Passenger Doors. Four (4) passenger entry doors are provided, one right and left at the forward and aft end of the passenger compartments. These doors are pneumatically operated for passenger loading and egress. Safety features are provided which preclude the doors closing on and injuring personnel, and which provide annunciation of a "door ajar" condition when underway.

3.5.4.4 **Emergency Escape Exits.** In addition to the above listed doors, four (4) side hatches (140 passenger) and two (2) side hatches (45 passenger), four (4) overhead hatches and a forward and aft door are provided for emergency egress.

3.5.5 **Equipment Compartments.** There will be equipment compartments located throughout the vehicle that can be easily accessed during operation by the crew.

3.6 **CONTROL SURFACES.**

3.6.1 **Forward Control Surfaces.**

3.6.1.1 **Forward Fin.** The forward fin supports the forward rudder and the forward horizontal stabilator. It is of graphite epoxy construction.

3.6.1.2 **Forward Rudder.** The forward rudder is attached to the forward fin. This surface is driven by an electro-mechanical actuator to provide yaw control forces.

3.6.1.3 **Forward Horizontal Stabilator.** The horizontal forward stabilators are attached to the forward fin. These surfaces are driven by an electro-mechanical actuators to provide pitch control forces, and roll forces when operated differentially.

3.6.2 **Aft Control Surfaces.**

3.6.2.1 **Aft Fins.** Two vertical aft fins provide yaw stabilization and mount the aft rudders and elevons. These surfaces are of graphite epoxy construction.

3.6.2.2 **Aft Rudders.** A movable rudder control surface is mounted to each aft vertical fin. These surfaces are driven by an electro-mechanical actuator to provide yaw control forces.

3.6.2.3 **Aft Elevons.** The aft horizontal control surface is split into a left and right half which can move independent of the other. They are mounted between the aft fins at the top. These surfaces are driven by electro-mechanical actuators to provide pitch and roll control forces

3.7 **DELETED**

3.8 **LANDING GEAR**

3.8.1 **Description.** The landing gear utilizes retractable skids and shock absorbers. Both a normal set of skids and a set of emergency braking skids are provided. The normal landing gear

3.8.1 Description. (Continued)

consists of four (4) retractable skid assemblies, two (2) forward and two (2) aft. These skids are provided with low friction, air lubricated pads. These pads are supplied with pressurized air through a system of manifolds. The skids contain features which allow them to conform to both the curved guideway, as well as a flat surface as might be encountered in a maintenance dock area.

In addition to the normal skids described above, a separate set of retractable skids are provided for emergency braking. These skids have high friction characteristics which permit deceleration of 0.5g. They are similar to the normal skid system but do not contain the air bearing features.

The retraction system is capable of lifting the vehicle above normal levitation heights so that the weight of the vehicle is carried by the skids. The landing gear is shown in Figures A-5 and A-6.

3.8.2 **Arrangement.** The forward emergency skids are located just aft of the forward levitation module and the aft emergency skids are located just forward of the aft levitation module. The normal skid system is located between the forward and aft emergency skids. Each skid assembly is attached to a trailing link shock absorbing mechanism which employs an oleo-pneumatic shock absorber. The trailing link mechanism is retractable and includes two (2) extended positions, one (1) for operation in the guideway and the other for supporting the vehicle on a flat surface such as a maintenance dock or other similar facility.

3.8.3 **Shock Absorber.** An oleo-pneumatic shock absorber operates in concert with the trailing link mechanism to provide approximately a two-inch (.05 m) stroke from zero load to fully compressed (strut bottomed out). Means of servicing the strut shall be provided.

3.8.4 Retracting, Extending and Locking.

3.8.4.1 **Extension/Retraction Features.** The extension system shall be capable of extending the gear when the weight of the vehicle is on the skids. The normal extension system is hydraulic and shall include provisions to coordinate the motions of the four (4) landing gear so that the vehicle is not tilted excessively during extension and retraction. The emergency skid system utilizes accumulators to provide a three-second extension to the down/locked position during emergencies. Motions of the four (4) emergency skids are not coordinated.

3.8.4.2 Locking. Features to lock the gear in any of its normal positions shall be provided.

3.8.5 Skid Special Features.

3.8.5.1 Articulation of Skid. The skids incorporate features which allows the weight to be evenly distributed while allowing them to conform to either a curved guideway or a flat surface. These features are shown in Figure A-5.

3.8.5.2 Skid Pad Features. High friction pads provide approximately a 0.5 coefficient of friction. The low friction air-lubricated pads provide for a 0.05 coefficient of friction.

3.9 FLIGHT CONTROL SYSTEMS

3.9.1 Description. The flight controls consist of the systems required to operate the forward and aft control surfaces. This system consists of sensors which determine vehicle velocity, linear and angular accelerations in all three (3) axis, a flight control computer which generates control commands in response to the sensor inputs and an actuation system in which electro-mechanical actuators respond to the flight control computer commands in order to dampen the vehicle response to disturbances in all six (6) degrees of freedom. The aerodynamic control system purpose is to achieve the desired ride qualities. The elements of this system are further described below.

3.9.1.1 Pitch/Roll Control System. The pitch control system actuates the forward and the aft stabilators (2). This system is capable of generating vertical forces by combining an up (or down) force at both ends of the vehicle as well as pitch motions by combining opposite forces at the forward and aft end of the vehicle. These motions all involve symmetrical left/right motions of the control surfaces. Roll control forces area generated by unsymmetrical left/right motions of both the front or rear surfaces.

These surfaces are controlled by irreversible dual load path electro-mechanical actuators.

3.9.1.2 Yaw Control System. The yaw control system actuates the forward (1) and aft rudders (2). This system is capable of generating side forces by combining symmetrical front/back deflections, or yawing forces by combining unsymmetrical front/back deflections. These surfaces are controlled by irreversible, dual load path electro-mechanical actuators.

3.10 PROPULSION AND LEVITATION

3.10.1 Description. The Magneplane is propelled by a synchronous linear electric motor concept in which super-conductive magnets in the vehicle interact with the guideway to both levitate and propel the vehicle. This specification will only deal with the vehicle elements of this system.

3.10.2 Magnet Installation. The magnet installation is shown in Figure A-7 and is further described below.

3.10.2.1 Propulsion Magnets. The super-conducting propulsion magnets are located in a forward and aft module along the centerline of the bottom of the vehicle. They are supported by a cryogenic system which supplies liquid helium to maintain the low temperatures required for super-conductivity. Provisions for disconnecting and purging the cryogenic system for maintenance are provided. Provisions for initiating super-conductivity are provided. These magnets interact with propulsion windings in the guideway to propel and brake the vehicle.

3.10.2.2 Levitation Magnets. The super-conducting levitation magnets are located in a forward and aft module on each side of the propulsion magnets. They are supported by a cryogenic system which supplies liquid helium to maintain the low temperatures required for super-conductivity. Provisions for disconnecting the cryogenic system for maintenance are provided. Provisions for initiating super-conductivity are provided. These magnets interact with the guideway to levitate the vehicle approximately six (6) inches (.15 m) above the guideway when underway.

3.10.2.3 Magnet Suspension. The magnets are hard-mounted to the airframe, by means of a suitable combination of mounts, fasteners and the vehicle keel structure. Provisions for replacement of the magnets are provided.

3.10.2.4 Cryogenic System. A cryogenic system is provided to supply the magnets with liquid helium in order to sustain super-conductivity. The cryogenic system consists of a compressor, insulated tank, heat exchanger (with electric fan), and gas return tank. These elements are located at the aft end of the vehicle. Lines connect the system to the magnets. Appropriate annunciation (including tank quantity) is provided to the vehicle operator to monitor performance of the system. Appropriate insulation is provided to the liquid helium supply lines. Provisions for recharging the system is provided. Provisions for lubricating the compressor as well as replenishing and cooling the lubricating oil are provided.

3.10.3 ON-BOARD CONTROL EQUIPMENT.

3.10.3.1 Propulsion/Speed Control. Propulsion/speed control - the speed of the vehicle is controlled by the wayside control system external to the vehicle. This system monitors vehicle position and positions the traveling power wave in the guideway to accelerate the vehicle and maintain the "command" speed. The vehicle equipment compares the actual speed to the command speed and generates an error signal which is transmitted to the wayside control system so that the traveling wave can accelerate or decelerate the vehicle to the "command" speed. This equipment is mounted in the vehicle's forward equipment bay.

3.10.3.2 Deceleration/Braking Control - (Normal). The normal braking of the vehicle is controlled by the wayside controller in a manner similar to the propulsion speed control described above. Normal braking is accomplished when a "command" speed is generated which is less than the actual vehicle speed. The traveling wave is positioned to decelerate the vehicle. This function is provided by the same equipment described above under the Propulsion/Speed Control.

3.10.3.3 Deceleration/Braking Control (Emergency). When an emergency situation is detected, either by the control system or the operator, the vehicle emergency skids are extended which provide a nominal 0.5g deceleration, independent of the magnetic propulsion/braking.

3.11 OPERATOR DISPLAYS

The vehicle operator station is equipped with appropriate displays and warning annunciations to permit the status of the vehicle systems to be determined and for the operator to be alerted to abnormal situations in time to take appropriate actions.

3.11.1 Warning Annunciations. The vehicle control system senses abnormal situations when regular vehicle parameters fall outside a normal band. A warning message is then transmitted to the wayside control system, as well as being displayed to the vehicle operator. Parameters subject to warning attention include: command speed versus actual speed, cryogenic system parameters, magnetic field strength, vehicle height above guideway, normal and emergency skid positions, ice detection, unlocked doors and other appropriate parameters.

3.11.2 Advisory Displays. The vehicle operator station is equipped with appropriate advisory displays to assist in normal operation. This category of display includes clocks, position along route, schedule status, etc.

3.12 AVIONICS.

3.12.1 Description. The avionics installations shall take into account requirements for lightning strike protection, Electro-magnetic Capability (EMC), security requirements, etc. Avionics installations include the vehicle control equipment, and other devices as outlined in the following:

3.12.2 Emergency Locator Transmitter (ELT)

3.12.3 Cockpit Voice Recorder - (CVR)

3.12.4 Crash Survivable Flight Data Recorder (FDR)

3.12.5 Flight Control Computer. This system controls the moveable control surfaces to achieve the desired ride qualities. This system is integrated with the vehicle speed control and braking systems.

3.12.6 Vehicle Position Sensing - (Normal). A system is installed along the guideway to allow the local and global control system to sense the position of each vehicle. The vehicle contains equipment to interface with this system.

3.12.7 Vehicle Position Sensing - (Back-up). A global positioning system shall be installed as a back-up for determining the vehicle position within the system. This system shall be capable of providing information for position displays of position in the vehicle as well as transmitting vehicle position to the control center.

3.12.8 Communications. The following types and quantities of transceivers shall be considered at this time for sizing requirements. The equipment shall be operable from the operator station crew members.

-(2) - Very High Frequency (VHF/AM)
-(1) - Very High Frequency (VHF/FM)
-(1) - Ultra High Frequency (UHF)
-Intercom - External and Internal jacks shall be provided.
-Public Address (PA) System
-Secure Voice

3.13 HYDRAULIC SYSTEM

3.13.1 Description. The hydraulic system for the Magneplane vehicle consists of two (2) independent 3,000 psi ($2x10^7 N/m^2$) systems, one located at the forward and aft end of the vehicle. This system provides hydraulic power for the extension and retraction of the landing gear skid system. Under normal operation, the forward system operated the forward skids and the aft system the aft skids. The system is configured so that upon failure of either system, the remaining system may operate both forward and aft skid systems, although at a slower rate than normal.

The system shall provide annunciation of failure and isolation of any sub-system failure that might cause an excessive loss of hydraulic fluid. Filters, accumulators and reservoirs shall be located to allow routine servicing without the use of special equipment or unusual access. The components of the system are further described below.

3.13.1.1 Hydraulic Pumps. The hydraulic pumps are electrically-driven and respond on demand to maintain the system accumulators at a nominal 3,000 psi ($2x10^7 N/m^2$). Two pumps are provided, one at each end of the vehicle.

3.13.1.2 Accumulators (Normal Operation). Each hydraulic pump is provided with an accumulator.

3.13.1.3 Accumulators (Emergency). Each hydraulic pump is paired with an emergency accumulator which is capable of fully extending the emergency skids on that end of the vehicle without the hydraulic pump.

3.13.1.4 Annunciation. Appropriate annunciation shall be provided to the vehicle operator to permit him to determine the status of the system and to deal with any system failures.

3.13.1.5 Plumbing Components. All lines, valves, etc., shall be designed to prevent corrosion, and shall be located so that any rupture would not cause a hazard.

3.13.2 Summary of Actuated Items. The following systems utilize hydraulic pressure from the above system:

Landing Gear Extension System

3.14 PNEUMATIC SYSTEM

3.14.1 Description. The Magneplane pneumatic system supplies compressed air by means of an electrically-driven air compressor for operation of pneumatically powered systems on the vehicle, primarily the air lubricated low friction skids. The system includes air tanks, gaging and annunciation, supply lines and electrical interface. The system provides low pressure air (50 - 70 psi)(3.4-4.8x10^5N/m^2), which is further reduced in pressure for specific applications by pressure regulators. Systems operated by compressed air include: the air bearing skids, passenger doors and pneumatic control surface deice boots. Specific system elements are described below:

3.14.1.1 Compressor. An electrically-driven compressor is provided. This system provides regulated low pressure compressed air at 50 to 70 psi (3.4-4.8x10^5N/m^2).

3.14.1.2 Compressed Air Tanks. Suitable tanks for accumulating compressed air are provided. Suitable protection is provided to protect the passengers and vehicle from the rupture of a tank.

3.14.1.3 Moisture Removal. Provisions for removal of excessive moisture from the compressed air supply shall be provided.

3.14.1.4 Gaging and Annunciation. The vehicle operator will be provided with appropriate gaging and annunciation to ensure the system's proper operation. These shall include system pressure, overpressure annunciation, etc.

3.14.2 Summary of Actuated Items. The following systems utilize bleed/pressurized air from the above system:

Air Lubricated Skids
Passenger Doors
Pneumatic Surface Deice
Passenger Loading Ramp (if required)

3.15 ELECTRICAL SYSTEMS (SUBTITLE)

3.15.1 Description. The electrical system includes all components required for the generation, conversion, storage, distribution, control and protection of electrical power. The electrical system shall be sized for the electrical loads plus a reserve for future growth. The electrical power distribution and control system configuration shall provide multiple (redundant) power paths for all mission critical functions, while maintaining the simplest bus arrangement for reliable power distribution.

3.15.2 Electrical Power. As described in the following sub-paragraphs, primary electrical power is provided by power from the guideway. The electrical system takes power from the guideway by means of a set of induction coils along the bottom centerline of the vehicle. This power is then converted to power the utility functions onboard the vehicle. Power is also stored in a series of storage batteries located in the front of the vehicle.

3.15.3 Pick-up Induction Coils. The pick-up induction coils are located along the bottom centerline of the vehicle interact with magnetic fields induced by the powered windings in the guideway to create a variable frequency alternating current in the pick-up coils. This current is then available for conversion to 270 volt pulsating direct current for charging the batteries and powering the battery bus.

3.15.4 Storage Batteries. A bank of 24 volt DC aircraft-type batteries is located at the forward end of the vehicle and is capable of supplying all vehicle power needs for thirty (30) minutes (1800 s) during an emergency. These batteries are connected in series to provide a nominal 264 volt DC bus voltage.

3.15.5 Power Conversion. DC power from the batteries or from the battery charging current is converted to three (3) phase 115 volt 400 hz AC current to power the onboard avionics and other appropriate systems.

3.15.6 External Power. The vehicle shall be equipped to accept electrical power from an external source.

3.15.7 Summary of Actuated Items. The following items on the vehicle are electrically-actuated:

Flight Control Actuators
Avionics
Battery Charging
Lighting
Displays
Hydraulic Pumps
Air Compressors
Heating System
Cooling System
Ventilation System

3.15.8 Lighting. The vehicle shall be equipped for night operations and meet the lighting requirements of applicable FRA requirements.

3.16 VEHICLE NOISE CHARACTERISTICS

3.16.1 External Noise Characteristics. The vehicle external noise shall meet applicable EPA and FRA criteria.

3.16.2 Internal Noise Characteristics. The vehicle internal noise shall not exceed 85 dba at maximum operating speed.

3.17 FIRE PROTECTION

3.17.1 Materials. The cabin compartment interior materials shall comply with the burn requirements of FAR 25.

3.17.2 Fire Detection. Smoke detectors shall be installed in the vehicle with annunciation to the vehicle operator. In addition, critical elements of the vehicle systems shall be equipped for fire or over-temperature protection.

3.17.3 Fire Extinguisher. Hand-operated fire extinguishers shall be located at the forward and aft end of the passenger cabin.

3.18 PERSONNEL ACCOMMODATIONS

3.18.1 Crew Seating. Seating at the vehicle operator station shall be provided for the operator and an observer. Both seats shall be equipped with seat belts.

3.18.2 Passenger Seating. Seating shall be provided for 140 or 45 passengers. Each seat shall be equipped with a seat belt.

3.18.3 Sanitary Facilities. A separate Men's and Women's toilet shall be provided at the aft end of the passenger compartment.

3.19 ENVIRONMENTAL SYSTEM

3.19.1 Environmental Control System (ECS). The vehicle Environmental Control System (ECS) shall provide cooling, heating, ventilation, moisture and contamination control for the vehicle interior and avionic storage compartments. The ECS shall provide a nominal cabin pressurization level of 1/2 psi ($3.4x10^3 N/m^2$) when the vehicle is underway at high speed. This provides passenger protection from rapid pressure changes when the vehicle passes objects or structures.

3.19.1 Environmental Control System (ECS). (Continued)

The ECS System utilizes ram air for pressurization when the vehicle is underway. The incoming ram air is conditioned (heated or cooled) prior to being routed to the cabin through a system of ducts. Pressurization is controlled by outflow valves which regulate the amount of air exhausting from the cabin. Blowers provide air for heating and cooling when the vehicle is not in motion.

3.19.1.1 **Cooling.** The cooling system shall be capable of maintaining a 70 degree F (21°C)cabin temperature when underway, and a 80 degree F (27°C) cabin temperature when stopped (except for transient conditions), with an outside temperature of up to 120 degrees F (49°C). A trade-off study will be made during the design phase between a dedicated vapor cycle cooling system versus utilizing the cryogenic system for cooling.

3.19.1.2 **Heating.** The heating system shall have the capacity to maintain the cabin compartment at 80 degrees F (27°C) during both enroute and stopped conditions, at ambient temperatures down to -20 degrees F (-29°C). Resistance-type electrical heating elements are distributed in the ducting to provide hot air for heating.

3.19.1.3 **Air Distribution.** Conditioned air for the cabin compartment shall be circulated throughout the areas with both armrest level (heating) and overhead (cooling) ducts. Ventilation fans shall be used to mix and distribute the air at the proper velocities. The air exits from the cabin through below-floor ducts to the outflow valves at the front and rear of the cabin.

3.19.1.4 **Ventilation.** The normal ventilation and pressurization air inflow is by ram air when underway, with blowers supplying ventilation during stopped conditions. The vehicle is pressurized only when underway at speed, and is unpressurized when stopped. Ventilation flow rates shall be at least 20 CFM per occupant.

3.19.2 **Oxygen System.** An emergency oxygen mask and supply is provided at the forward and aft end of the cabin for medical emergency. This system requires two outlets, each capable of providing 100% oxygen at a flow rate of 35 liters per minute at 50 psi($3.4x10^5 N/m^2$) for up to 30 minutes (1800 s).

3.20 ALL WEATHER SYSTEMS

3.20.1 Description. The Magneplane operations include the requirements for operating in adverse weather conditions. In order for the vehicle to operate it must be capable of sustained operation into known icing conditions without undue performance penalties. Deice/anti-ice protection shall be required for the areas discussed in the sub-paragraphs to this section.

3.20.2 Airframe Protection. Airframe protection must prevent the buildup of ice on any portion of the vehicle that would adversely affect vehicle operation.

3.20.2.1 Surface Ice Protection. The Magneplane vehicle shall utilize deice boots on the leading edges of the vertical and the horizontal control surfaces.

Operation of the deice boot system is manually initiated by the crew when in icing conditions. The deice boot operating system shall detect ice conditions and the controller/timers shall have an automatic mode that cycles the boots at a given interval for flight in continuous icing conditions without crew action. Appropriate annunciation shall be provided to the vehicle operator.

3.20.2.2 Windows. The critical viewing areas of the operator's windows shall remain free of ice, frost, and fog; and have rain removal system wipers for transient and steady state portions of ground and underway operations.

3.20.2.3 Miscellaneous Anti-Ice Systems. The protection of any protruding probes and sensors shall be provided as required for continued operation in icing conditions.

3.21 SUPPORT PROVISIONS

3.21.1 Maintenance Manual. As described by contract, a complete maintenance manual(s) shall be provided to the user. Its contents shall provide all necessary information to maintain the vehicle's operational readiness.

3.21.2 Ground Handling. The Magneplane design shall utilize design features which allow the use of ordinary ground support equipment. This includes towing, jack points, tie-down features, ground support equipment and loose tools.

3.22 ROTOR BURST

3.22.1 Structure. The vehicle structure shall be designed so that the failure of rotating equipment elements shall not result in uncontained fragments causing a hazard to the vehicle or passengers.

3.22.2 Systems. All critical systems, such as flight controls, electrical, propulsion and braking controls, and cryogenic systems and hydraulic systems are routed such that rotor fragments cannot endanger the safety of the vehicle or passengers.

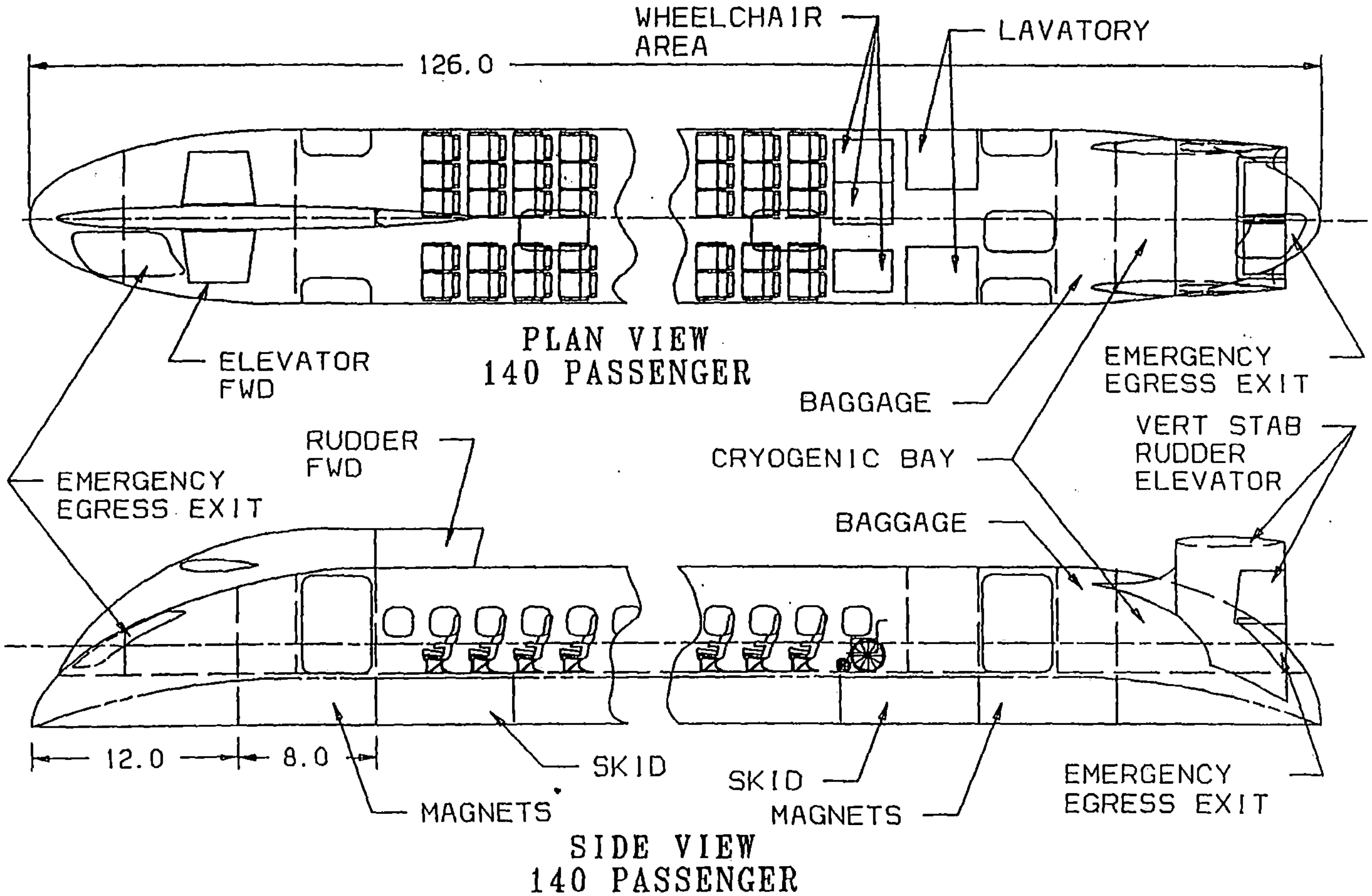


FIGURE A-1 - 140-Passenger Magneplane Vehicle

SECTION A

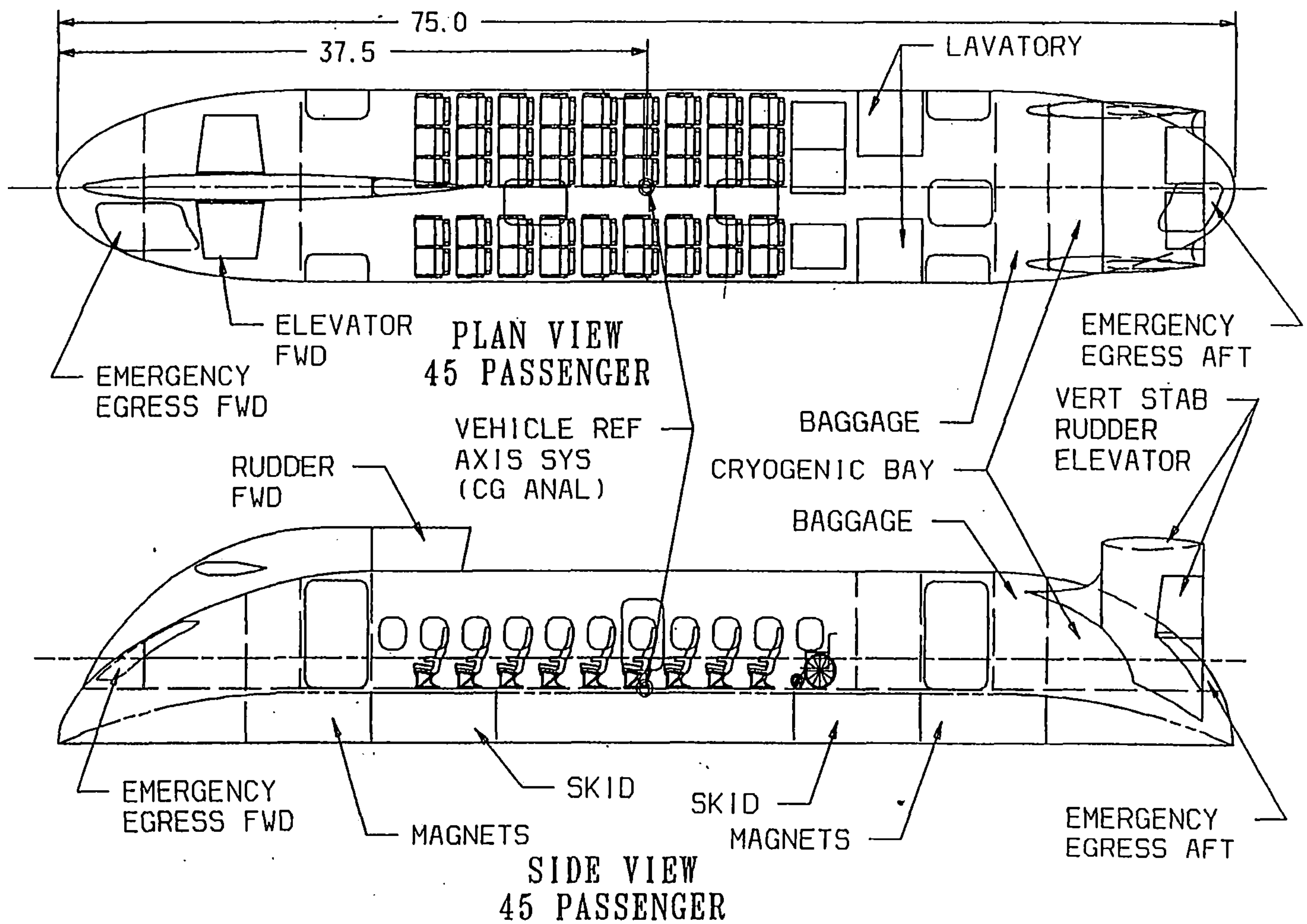


FIGURE A-2 - 45-Passenger Vehicle

SECTION A

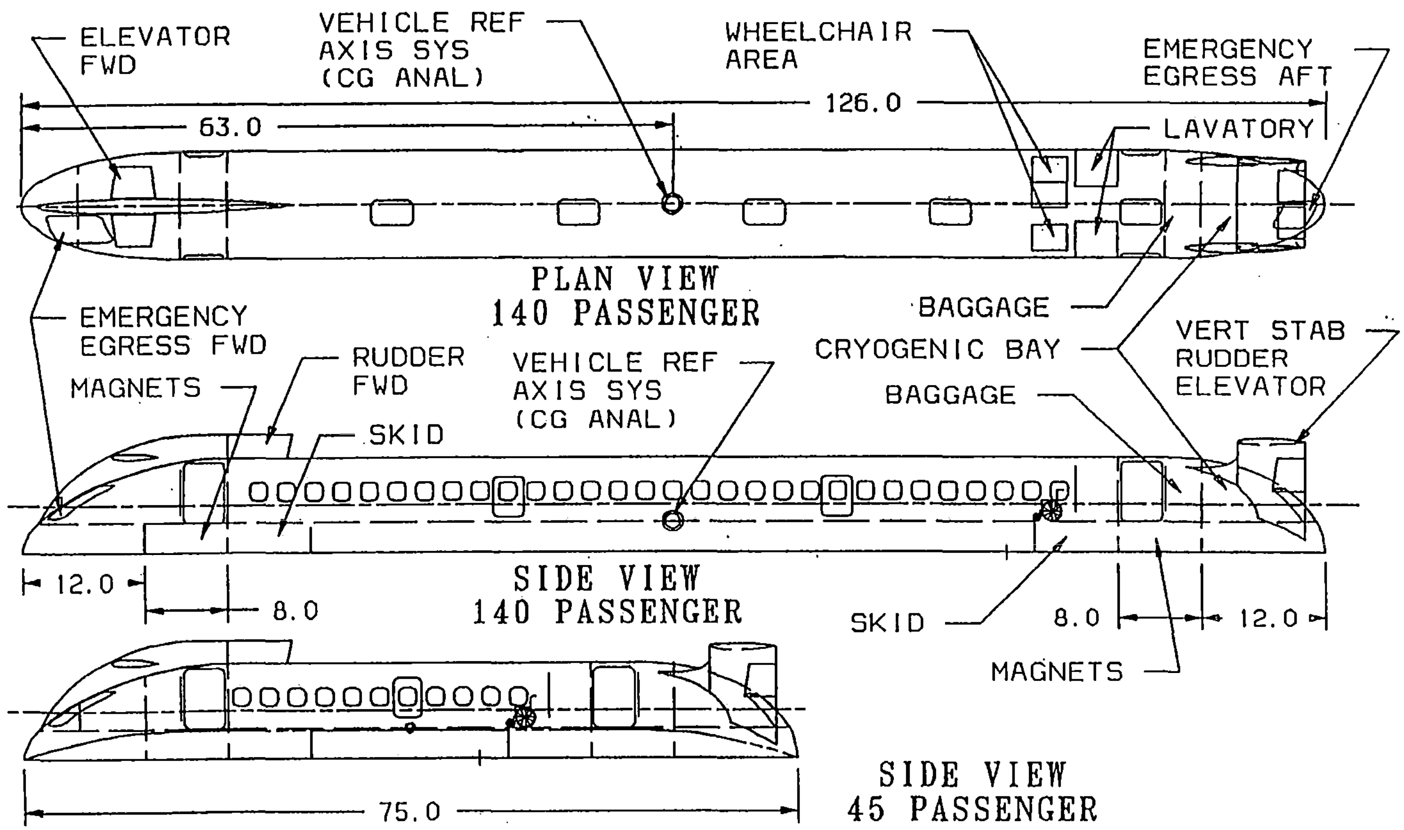


FIGURE A-3 - Emergency Exit Arrangements
SECTION A

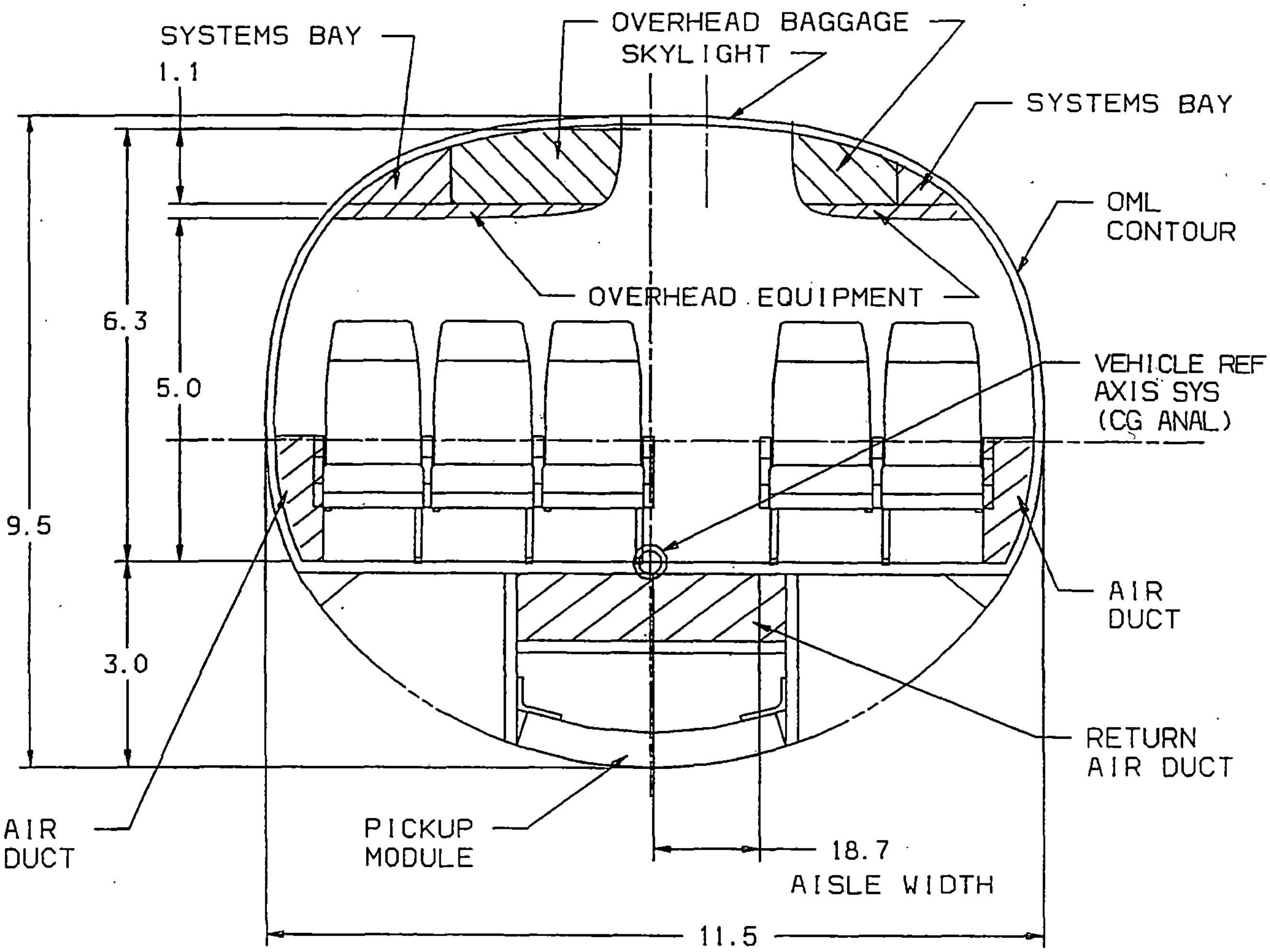


FIGURE A-4 - Air Distribution System
SECTION A

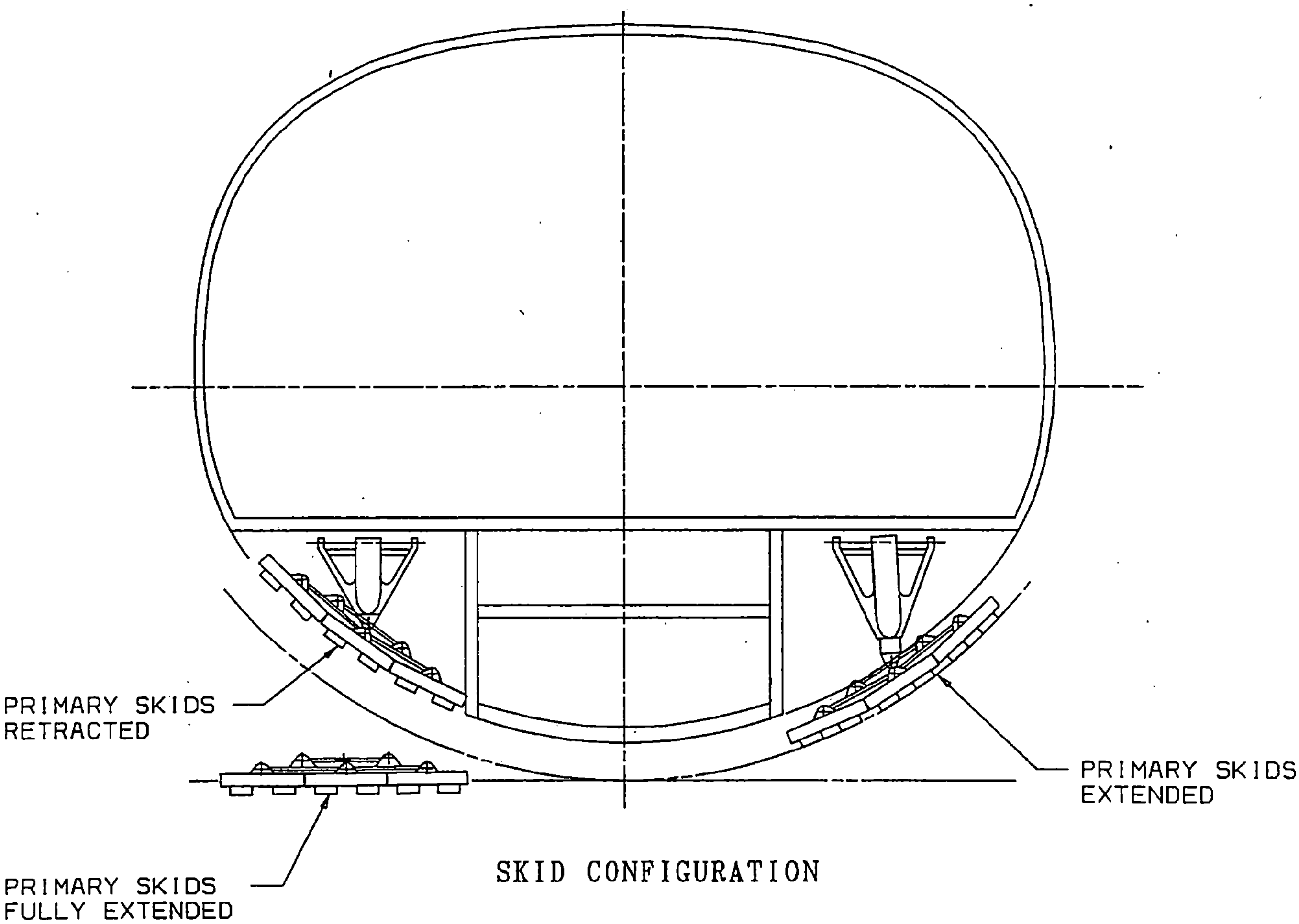


FIGURE A-5 - Ski-Skid Configuration
SECTION A

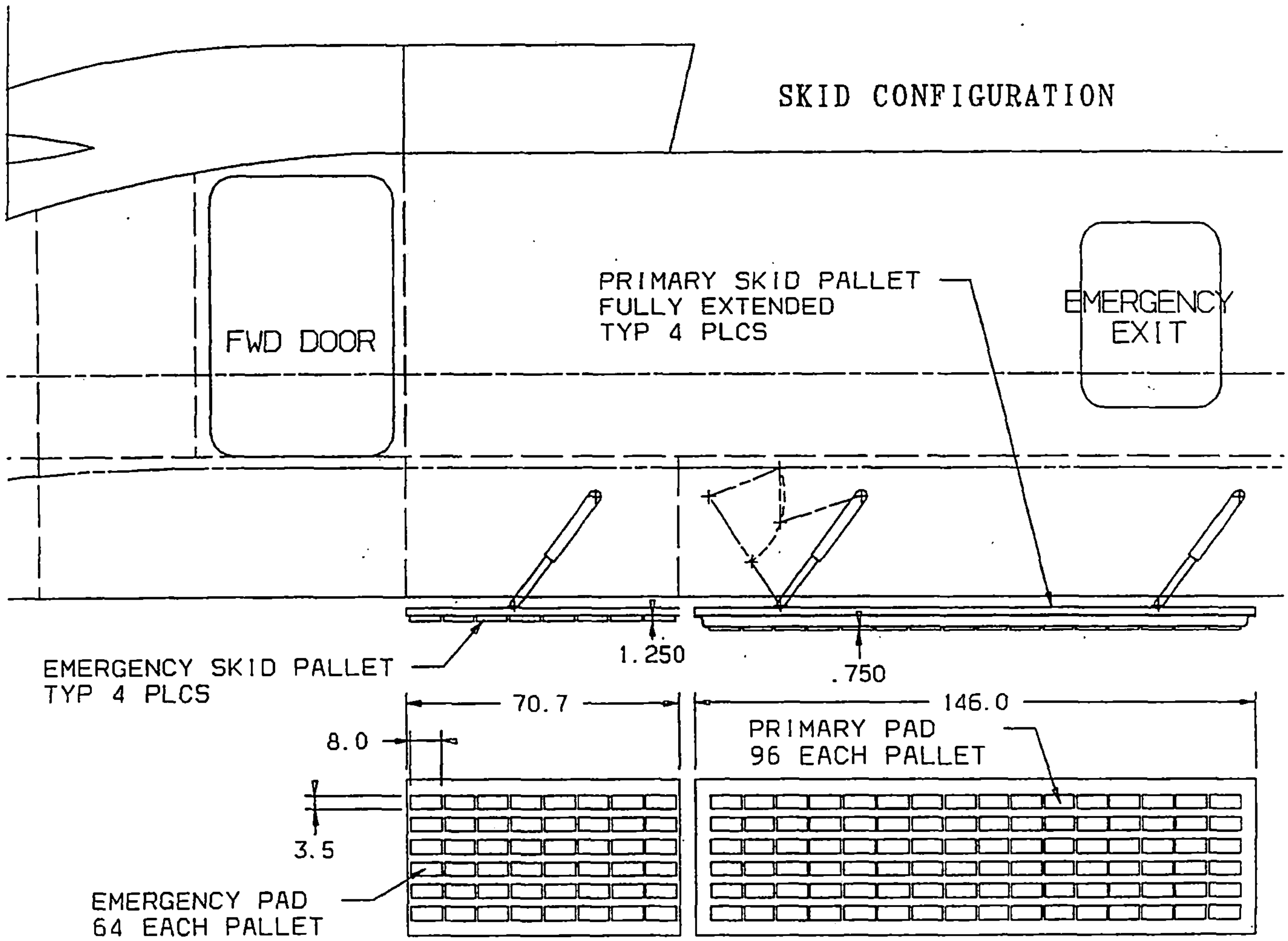


FIGURE A-6 - Anti-Friction Pneumatic Skid Pad
SECTION A

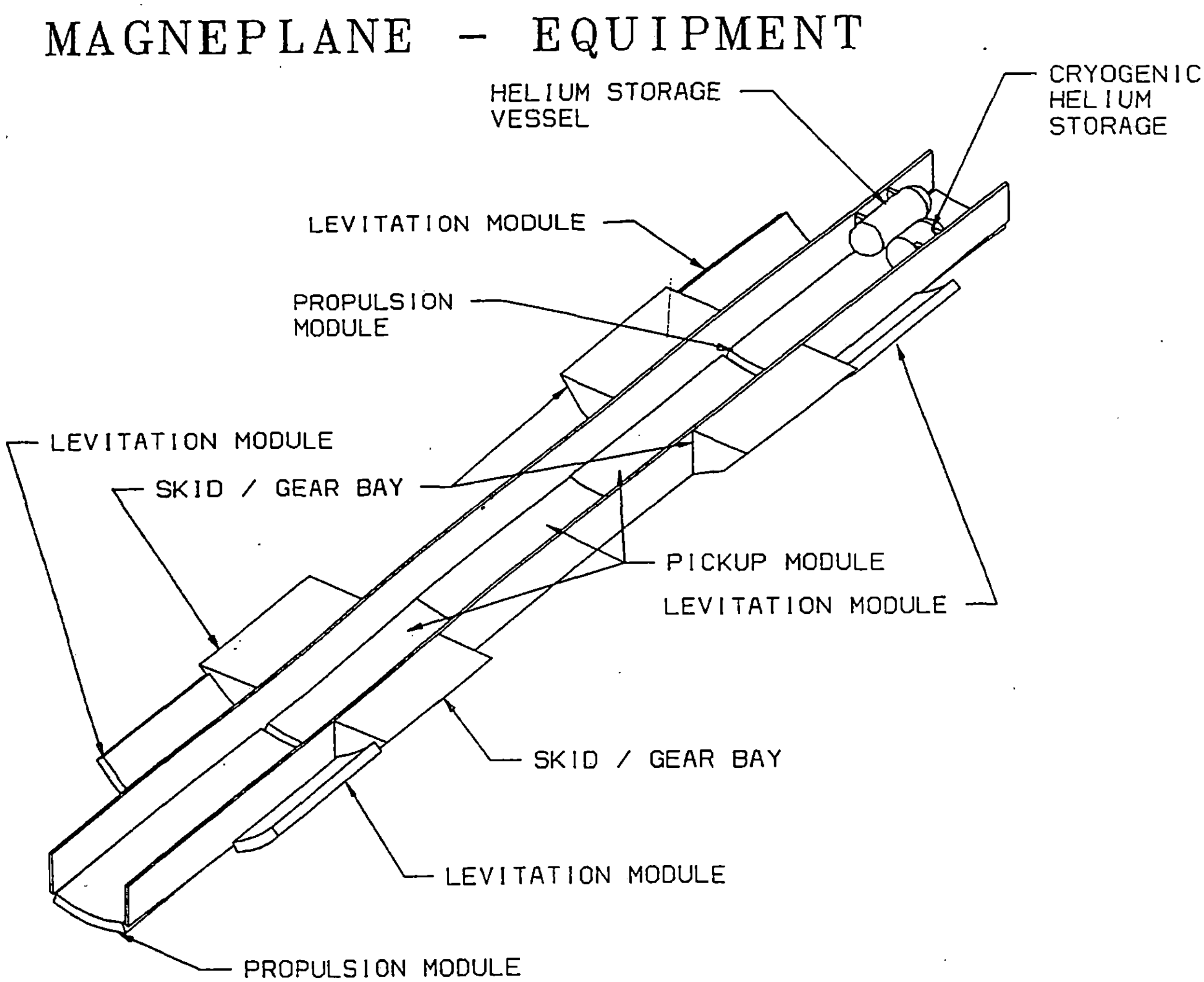


FIGURE A-7 - Cryogenics & Electrical (Under Floor)
SECTION A

SUPPLEMENT D
SECTION B

MAGNEPLANE FREIGHTER VERSION

INTRODUCTION: The Freighter version of the Magneplane is based on the passenger vehicle, modified to provide for carriage of palletized cargo.

The passenger seats are removed and the environmental system down-sized for the reduced ventilation, heating and cooling requirements. A swing out cargo hatch is provided in the center of the vehicle, with a floor mounted cargo handling system installed. This permits loading of pallets at the center of the vehicle, and moving them forward or aft into the cargo compartment. This variant is shown in Figure B-1. A weight estimate for this vehicle is shown in Figure B-2.

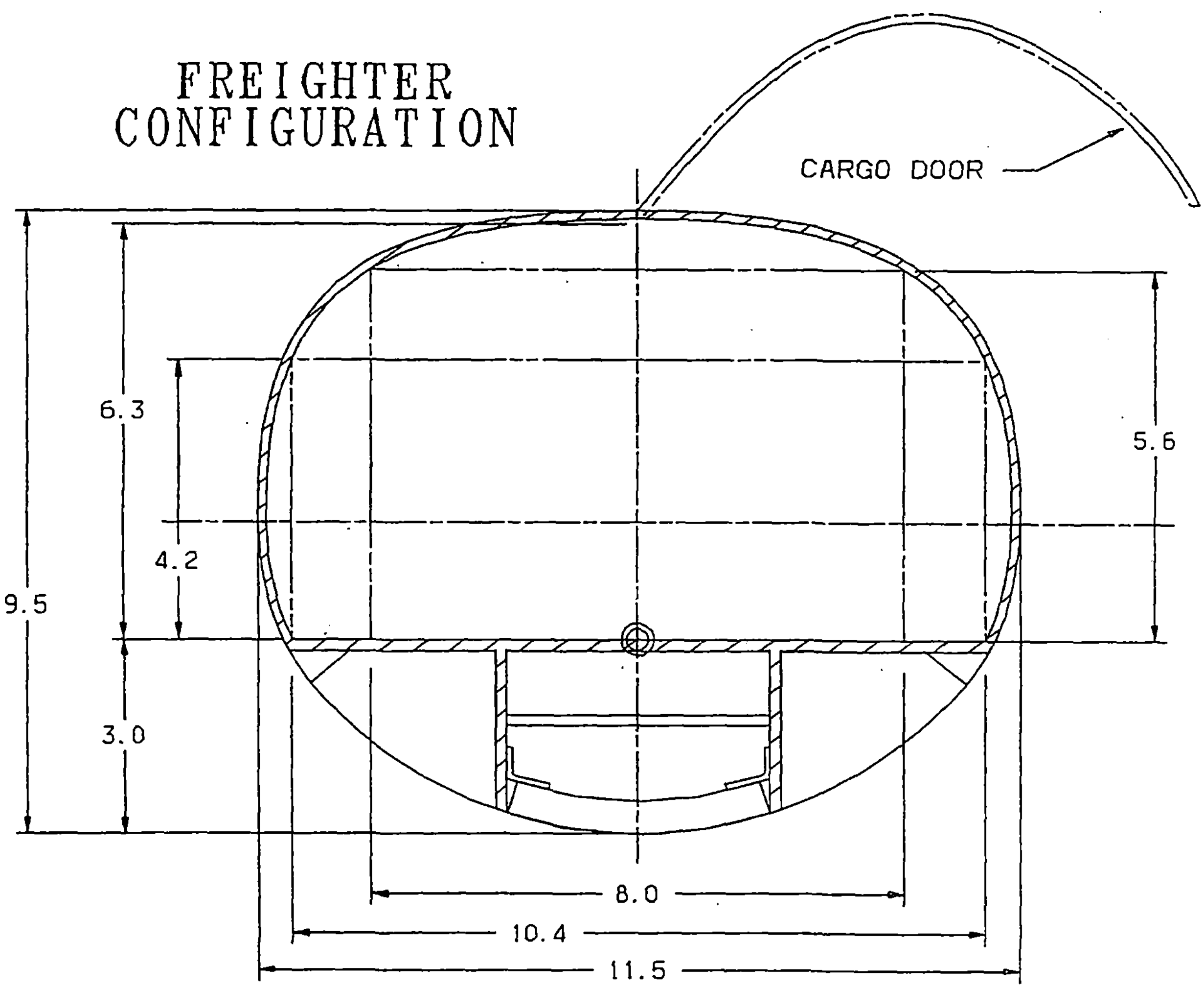


FIGURE B-1 - Magneplane Freighter Version
SECTION B

WEIGHT ESTIMATES
MAGNEPLANE FREIGHTER VERSION

Item No.	Item	Weight (Lbs)	Weight (KG)
1)	Airframe	18,200	8273
2)	Cargo	45,000	20454
3)	Interior	1,200	545
4)	Heating and Ventilation	1,200	545
5)	Levitation Magnets	5,673	2579
6)	Propulsion Magnets	6,747	3065
7)	Cryogenic System	2,000	909
8)	Suspension System	500	227
9)	Shielding	5,000	2273
10)	Skids/Landing Gear	1,440	654
11)	Controls/Stability Augmentation	4,000	1818
12)	Power Pick-up Coils	3,970	1804
13)	Battery	2,500	1136
14)	Handling Provisions	4,000	1818
15)	Miscellaneous	2,570	1168
TOTAL		105,000	47727

FIGURE B-2

SUPPLEMENT D
SECTION C

STABILITY AUGMENTATION SYSTEM

MAGLEV SYSTEM CONCEPT DEFINITION

1.0 INTRODUCTION: The stability augmentation system's basic function is to dampen out small disturbances which would tend to excite the vehicle structural modes, which would degrade ride qualities.

This system provides aerodynamic forces to counter these external disturbances which come from wind gusts or guideway perturbations. The stability augmentation system consists of a sensor package to sense disturbances, a system of aerodynamic control surfaces and actuators, and a computer system to invoke the control laws of the system. The system is integrated with the propulsion and braking systems and is a part of the Integrated Flight/Propulsion Control System (IFPS). A block diagram of the proposed system is shown in Figure C-1.

2.0 SENSORS: The stability augmentation system utilizes the following sensors:

1) **Linear Accelerometers:** Acceleration disturbances will be measured by linear accelerometers, using three orthogonal sensors. Sensors will be located in the forward and aft of the vehicle and displaced about the horizontal center line of the vehicle. The instruments are DC powered.

By measuring linear acceleration at two different locations in the vehicle, angular accelerations can be deduced.

2) **Air/Pressure Measurements:** Directional sensors will include Angle-of-Attack sensors for both pitch and yaw, and static and dynamic air pressure sensors.

3) **Height detection:** A photo-optical system detects vehicle height above guideway.

4) **Position/Velocity Sensing:** The vehicle senses position along the route, and track velocity, by a series of RF markers along the guideway.

5) **Linear Synchronous Motor Field Strength Sensors:** These sensors measure the strength of the magnetic fields and are used for the propulsion and braking functions.

All of the above sensors are dual, each set of sensors feeding separate channels. All sensor inputs are converted to digital form.

3.0 AERODYNAMIC CONTROL SURFACES: The Magneplane vehicle utilizes active aerodynamic control surfaces to provide stability augmentation and ride control. Two (2) horizontal stabilators are provided at the front of the vehicle, and two (2) horizontal control surfaces are provided at the aft end. These surfaces are capable of generating both pitch and roll control forces. A vertical yaw canard is provided on the forward end and double fins and rudders at the aft end, which are capable of generating yaw forces as well as side forces.

The vehicle control surfaces are shown in Figure C-2.

Control Surface Actuators: Control surface actuators are electro-mechanical, with each control surface actuated by dual actuators, each half tied to a separate control channel.

4.0 BRAKING CONTROL: Primary braking is by the synchronous motor. An emergency braking skid system is provided in the event of a failure of the primary system.

5.0 PROPULSION CONTROL: The vehicle is a passive element in the linear synchronous motor concept. The traveling wave is positioned on the guideway to accelerate or decelerate the vehicle. This system is integrated with the stability augmentation system so that propulsion accelerations are considered in the control laws.

6.0 FLIGHT CONTROL COMPUTER: Dual flight control computers act on sensory inputs, operator, and wayside controller commands, to control vehicle speed as well as to provide aerodynamic stabilization activities. These activities include positioning the vehicle roll orientation prior to entering a switch or turn, as well as providing compensating aerodynamic forces to oppose disturbances caused by wind affects, variations in levitation forces and guideway variations. These activities provide improved passenger ride qualities.

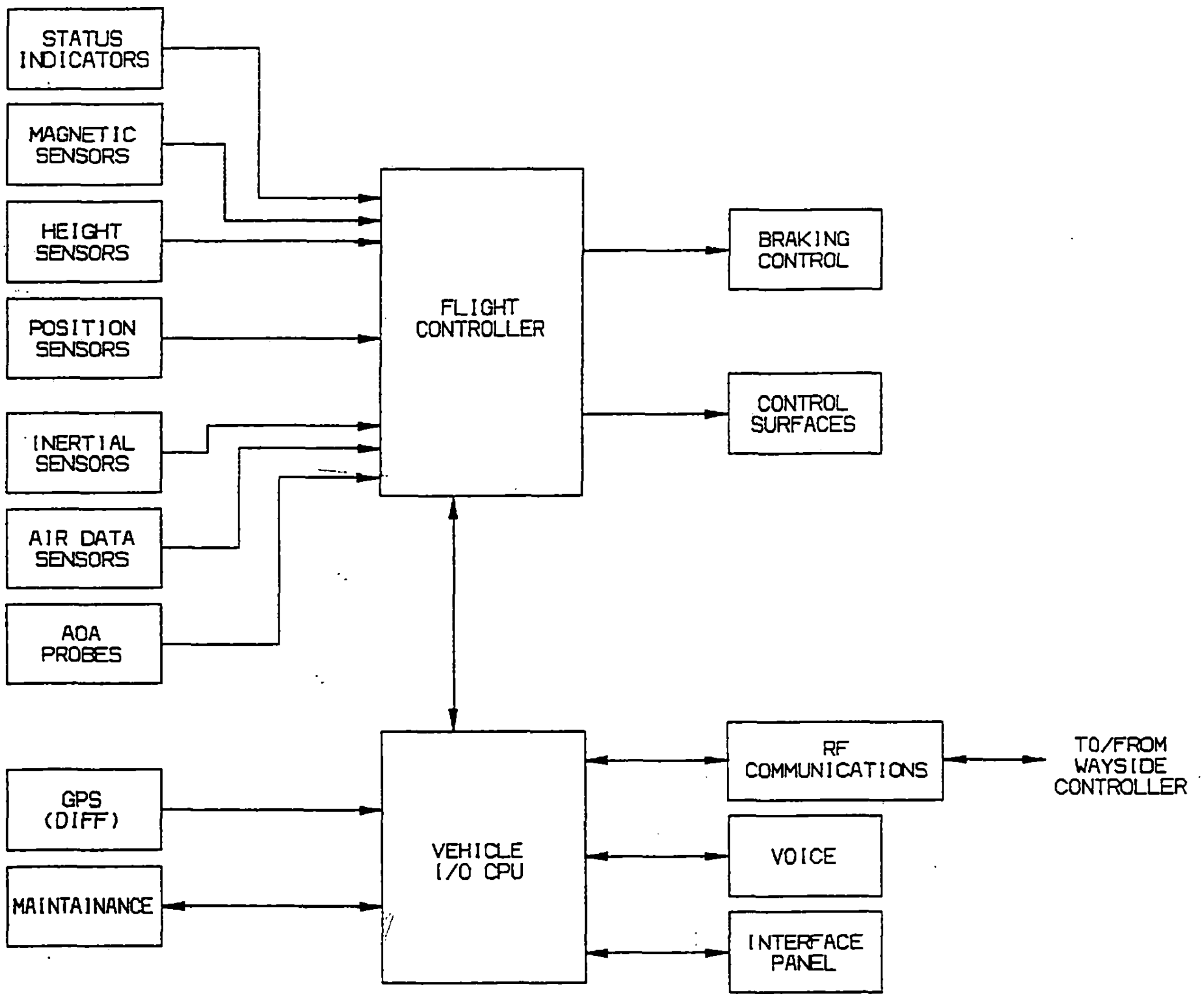


FIGURE C-1 - Stability Augmentation System
SECTION C

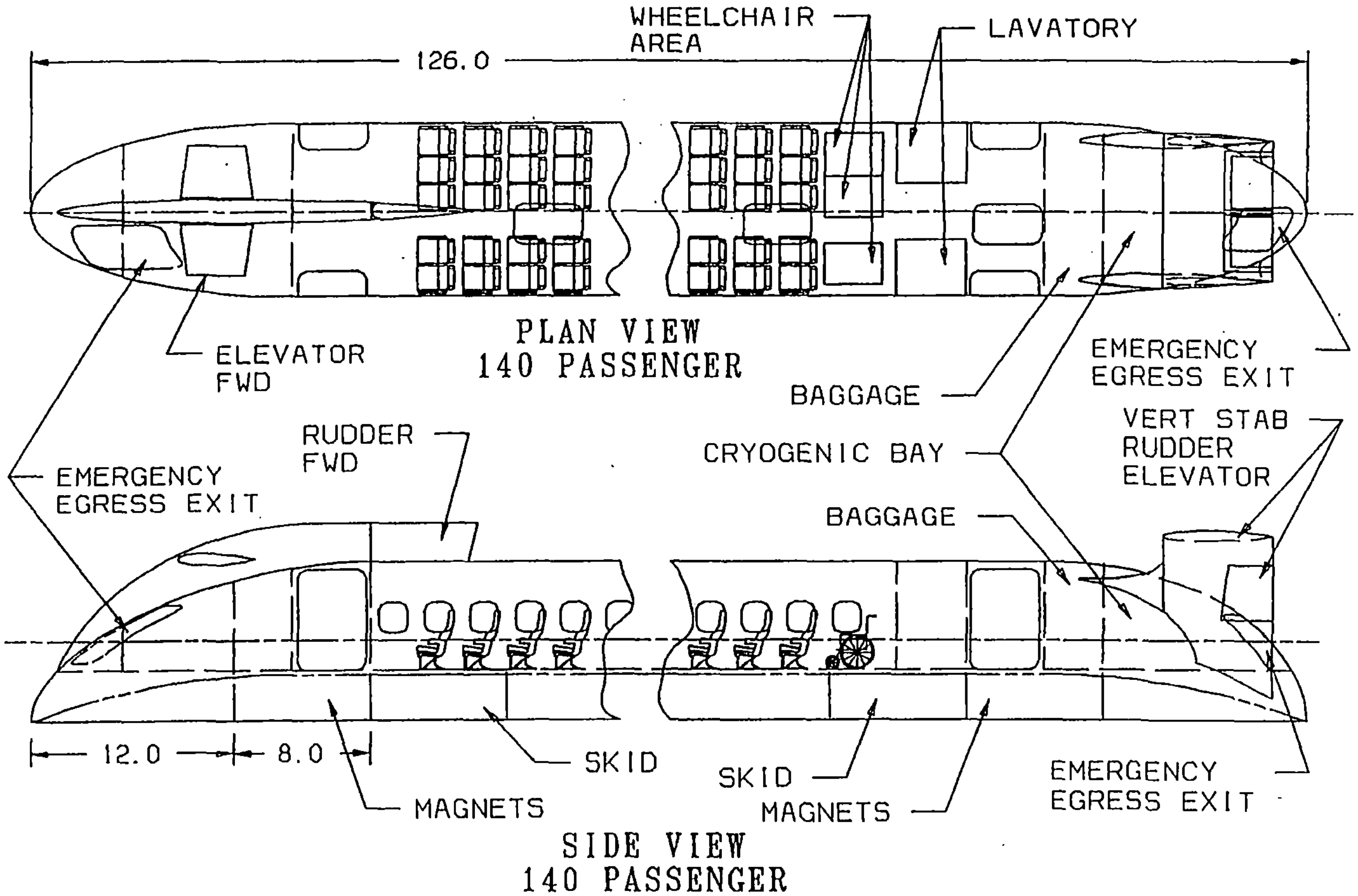


FIGURE C-2 - Side View - 140-Passenger Vehicle

SECTION C

SUPPLEMENT D
SECTION D

ALTERNATIVE WHEELED LANDING GEAR

MAGLEV SYSTEM CONCEPT DEFINITION

INTRODUCTION:

An alternative to the skid landing gear system was studied. This system would utilize conventional aircraft wheels and brakes. This system would require changes to the guideway configuration. The proposed configuration is shown in Figure D-1. This landing gear system would be a quad system with a landing gear on all four (4) corners of the vehicle. This landing gear system would incorporate an extension system to raise the vehicle above the levitation height.

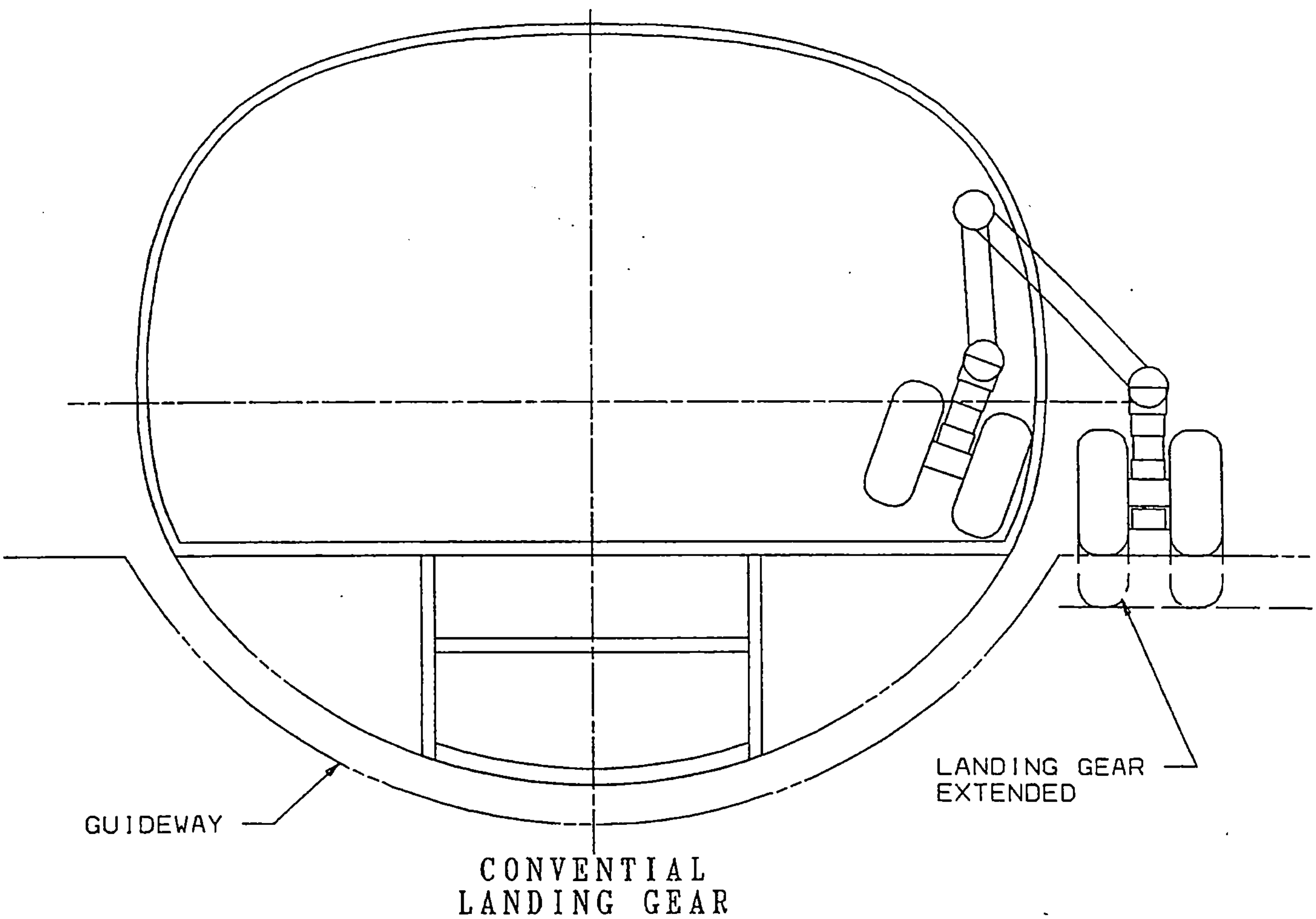


FIGURE D-1 - Alternate Wheeled Landing Gear
SECTION D

SUPPLEMENT D
SECTION E

MATERIAL TRADE STUDIES

MAGLEV SYSTEM CONCEPT DEFINITION

SECTION E

1.0 INTRODUCTION: This appendix describes a Materials Trade Study which was performed to determine the weight savings which could be achieved by utilization of advanced composite materials for the Magneplane airframe.

2.0 CONCLUSIONS: This study showed that the composite large vehicle could achieve weight savings of 7056 lbs (3207 KG). When the affect of this additional weight on the landing gear and levitation magnets were added, the total vehicle weight savings for a composite structure was 8,133 lbs (3697 KG) . This savings amounts to 28% weight savings for the fuselage structure. Since the structure is only a fraction of the total vehicle weight, the vehicle is 7% lighter when made of composite materials.

3.0 TRADE STUDY RATIONALE: The Materials Trade Study followed the following sequence: 1) Loads for the vehicle were generated, 2) A composite structure was defined which met the loads criteria, 3) An equivalent metallic structure was defined, 4) Weight comparisons were made and 5) a structural frequency was calculated for the resulting design.

4.0 LOADS: The structural loads were based on evaluation of 1g static loads based on the applied uploads of the levitation magnets which oppose the downloadings of the distributed and discrete masses of structure and components. This 1g static load was then multiplied by a 2.5g limit load plus a 50% safety factor. The design load factors are shown in Figure E-1 and the 1g internal static loadings are shown in Figure E-2. Maximum bending moment of over 760,000 lb/ft (1,000,000 N-m) occurs near the center of the vehicle. A design load for the fuselage of 600,000 lb/ft(814,650 N-m) was chosen for the generic fuselage section, since the entire fuselage would not be designed for the higher load. A maximum 1g shear of 37,000 lbs (16818 kg) occurs near the aft magnetic module. In addition to the above loads, a deflection requirement that the vehicle should not deflect more than 1 inch (.025 m)under the 1g static load was imposed.

STATIC DESIGN LOADS
(Normal Structural Load Factor)

Vertical Up Acceleration	(2.5)
Vertical Down Acceleration	(1.25)
Lateral Acceleration	(1.5)
Longitudinal Acceleration	(1.5)

Emergency Structural Loads (Applied at seats)
(Load Factor)

Vertical Up Acceleration	3.0 (Applied to seat, vehicle restrained).
Vertical Down Acceleration	6.0 " "
Rearward Acceleration	1.5 " "
Forward Acceleration	9.0 " "
Lateral Accel. (Airframe)	3.0 " "
Lateral Accel. (Seats & Attach)	4.0 " "

Note: The normal loads shall be assumed to be encountered in service. The emergency loads are assumed to be encountered only in an emergency and need not be addressed from a repeated load standpoint. The emergency loads are applicable for design of occupant protection features. Structural damage may be sustained when the emergency loads are applied as long as the failures do not jeopardize the occupants.

FIGURE E-1

MAGNEPLANE FUSELAGE INTERNAL LOADS IN LBS-FT

Fuselage Station (FT)	Total Shear (Lbs)	Total Moment (Lb-Ft)
0	0	0
2	317	317
4	635	1,270
6	1,687	2,856
8	2,005	6,548
10	2,322	10,875
12	2,639	15,835
14	8,009	21,432
16	-32,929	37,766
18	-32,611	-27,773
20	-32,294	-92,678
22	-31,031	-156,003
24	-27,268	-216,801
26	-26,005	-270,074
28	-24,742	-320,821
30	-23,356	-368,919
32	-21,971	-414,247
34	-20,585	-456,803
36	-19,200	-496,590
38	-17,815	-533,604
40	-16,429	-567,849
42	-15,044	-599,323
44	-13,658	-628,026
46	-12,273	-653,957
48	-10,888	-677,118
50	-9,502	-697,509
52	-8,117	-715,129
54	-6,732	-729,977
56	-5,346	-742,055
58	-3,961	-751,363
60	-2,575	-757,899
62	-1,190	-761,664

*Large Vehicle

FIGURE E-2

FIGURE E-2 (CONTINUED)
MAGNEPLANE FUSELAGE INTERNAL LOADS*In LBS-FT

Fuselage Station (Ft)	Total Shear (Lbs)	Total Mount (Lb-Ft)
64	195	-762,659
66	1,580	-760,883
68	2,966	-756,336
70	4,352	-749,019
72	5,736	-738,930
74	7,122	-726,071
76	8,508	-710,441
78	9,893	-692,040
80	11,278	-670,868
82	12,664	-646,929
84	14,049	-620,213
86	15,434	-590,728
88	16,820	-558,474
90	18,205	-523,448
92	19,591	-485,652
94	20,976	-445,085
96	22,362	-401,747
98	23,747	-355,638
100	25,010	-306,881
102	36,773	-255,597
104	37,090	-181,734
106	37,408	-107,235
108	37,725	-32,102
110	-5,274	43,665
112	-4,956	33,435
114	-4,639	23,839
116	-4,322	14,878
118	-2,004	6,551
120	-952	2,860
122	-635	1,273
124	-317	321
126	0	0

*Large Vehicle

MAGNEPLANE FUSELAGE INTERNAL LOADS*-N-m

Fuselage Station(m)	Total Shear (N)	Total Moment (N-m)
0	0	0
.61	1412	430
1.22	2829	1724
1.83	7515	3878
2.44	8931	7669
3.05	10343	14766
3.66	11756	21500
4.27	35676	29099
4.88	-146684	51277
5.49	-145267	-37709
6.10	-143855	-125833
6.71	-138229	-211813
7.32	-121467	-294361
7.92	-115840	-366692
8.53	-110214	-435594
9.14	-104040	-500899
9.75	-97871	-562433
10.36	-91697	-620223
10.97	-85527	-674244
11.58	-79358	-724500
12.19	-73184	-770996
12.80	-67014	-813730
13.41	-60840	-852701
14.02	-54671	-887909
14.63	-48501	-919356
15.24	-42327	-947042
15.85	-36158	-970965
16.46	-29988	-991125
17.07	-23814	-1007524
17.68	-17644	-1020162
18.29	-11470	-1029036
18.90	-5301	-1034148

*Large Vehicle

FIGURE E-2A

FIGURE E-2A (CONTINUED)

MAGNEPLANE FUSELAGE INTERNAL LOADS*N-m

Fuselage Station (m)	Total Shear (N)	Total Mount (N-m)
19.51	869	-1035499
20.12	7038	-1033088
20.73	13212	-1026914
21.34	19386	-1016979
21.95	25551	-1003281
22.56	31725	-985822
23.16	37899	-964600
23.77	44069	-939616
24.38	50239	-910870
24.99	56412	-878367
25.60	62582	-842093
26.21	68751	-802060
26.82	74925	-758267
27.43	81095	-710711
28.04	87269	-659393
28.65	93439	-604313
29.26	99613	-545471
29.87	105782	-482867
30.48	111408	-416667
31.09	163807	-347036
31.70	165219	-246749
32.31	166636	-145598
32.92	168048	-43586
33.53	-23493	59286
34.14	-22077	45396
34.75	-20665	32367
35.36	-19253	20201
35.97	-8927	8895
36.58	-4214	3883
37.19	-2829	1728
37.80	-1412	436
38.40	0	0

*Large Vehicle

STRUCTURAL DESIGN RESULTS: The composite structural design is shown in Figure E-3. The equivalent metallic design is shown in Figure E-4. Both designs were impacted by the imposition of the deflection criteria to a slight degree. In the case of the composite vehicle, the skin gages were increased on the top of the vehicle from .05" to .06" (1.3-1.5 mm) and the keel beam cap was added to meet the deflection criteria. Both versions, composite and metallic have a free-free vertical fuselage bending frequency of 4.3 hz. If a higher frequency were to be required, additional material would need to be added. This was considered beyond the scope of this study. The weight of the composite fuselage is 99 lbs per foot (148 kg/m) and the metallic fuselage is 155 lbs per foot(230 kg/m). These results were used to estimate the total vehicle weights which are shown in Figure E-5. The total vehicle weights reflect the need for a heavier landing gear, levitation magnets and shielding for the heavier metallic vehicle.

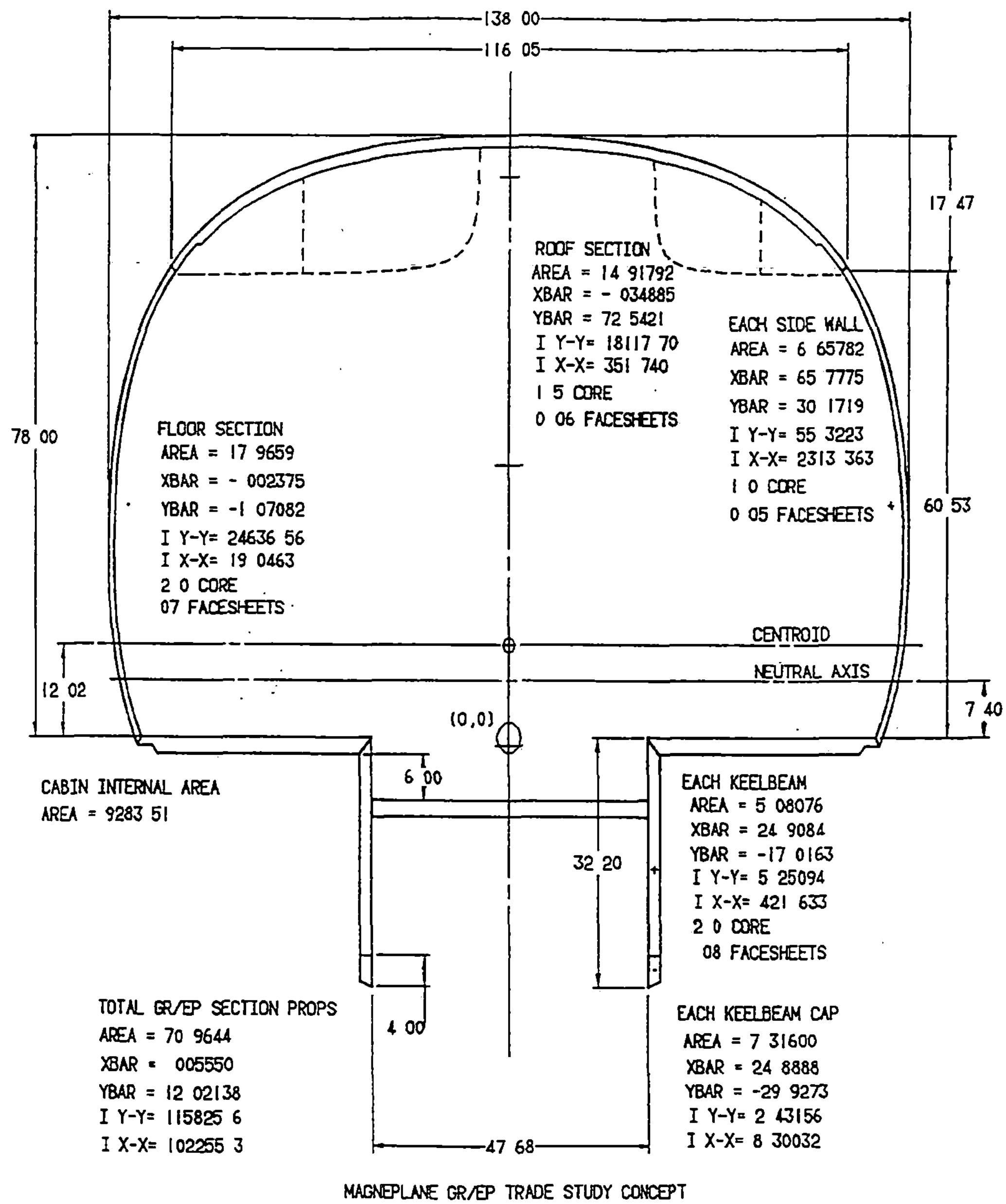


Figure E-3 Magneplane GR/EP trade study concept

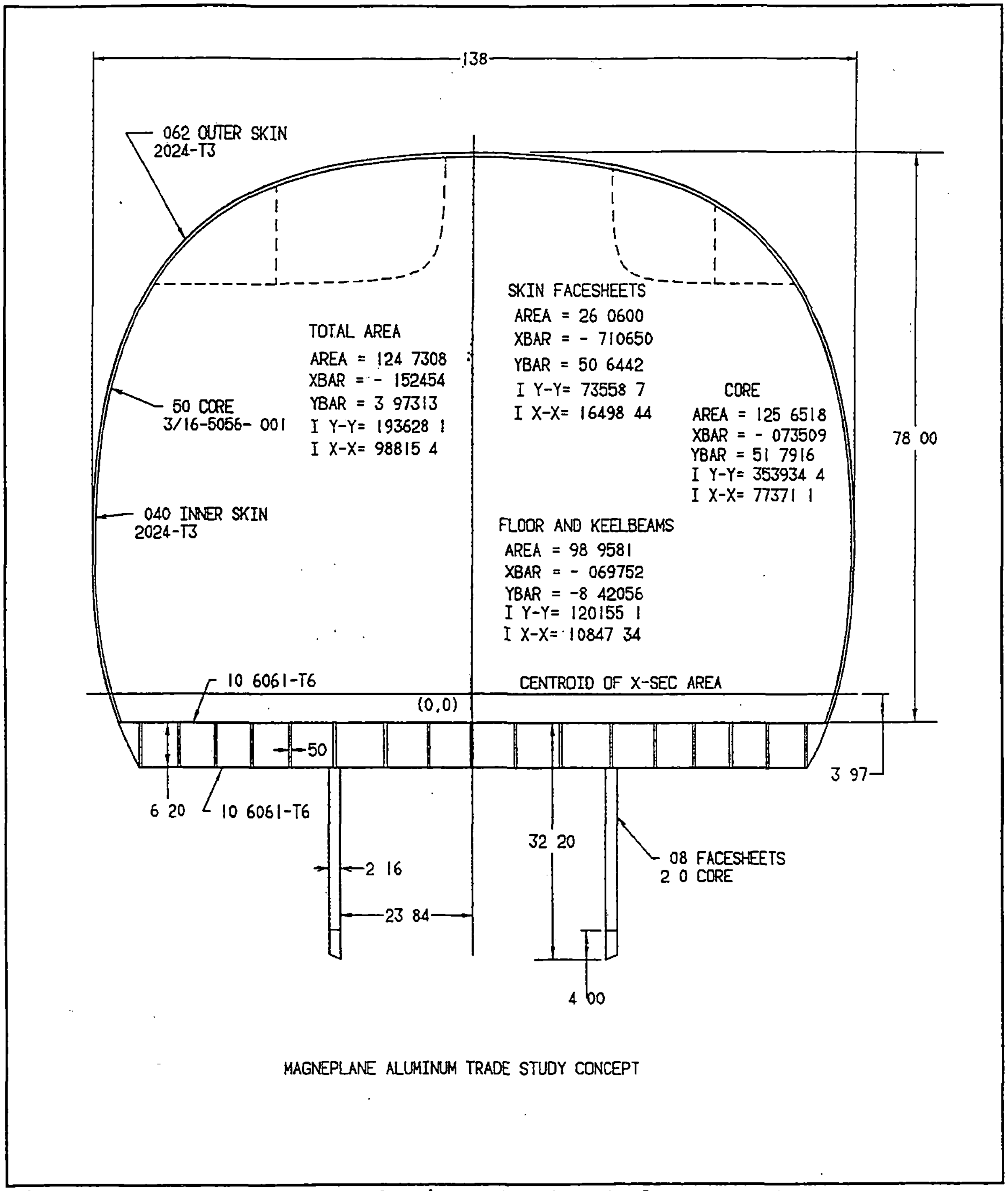


Figure E-4 Magneplane aluminum trade study concept

LARGE VEHICLE WEIGHT COMPARISONS - (LBS)

	Composite	Metallic
Structure (Shell)	12474	19530
Structure (Other)	5570	5570
Passengers	30690	30690
Freight	8000	8000
Interior(1)	4480	4480
Heating/Air Conditioning (1)	3600	3600
Magnets(2)	12420	12972
Cryogenics(1)	2000	2000
Suspension	500	500
Shielding(2)	7518	7743
Skids/Landing Gear(2)	5000	5000
Controls/Stability Augmentation	1440	1440
Power Generation, etc.	9357	9357
Miscellaneous/Contingency	1951	1951
TOTAL	105,000	113,133

NOTE: (1) Weight driven by number of passengers.
(2) Weight driven by gross weight.

SUMMARY:

- Structural composite weight savings - 7056 lbs (28% of metallic fuselage structural weight).
- Total vehicle composite weight savings - 8133 lbs (7% of metallic vehicle total weight).

FIGURE E-5

LARGE VEHICLE WEIGHT COMPARISONS - (KGS)

	Composite	Metallic
Structure (Shell)	5670	8877
Structure (Other)	2532	2532
Passengers	13950	13950
Freight	3636	3636
Interior(1)	2036	2036
Heating/Air Conditioning (1)	1636	1636
Magnets(2)	5654	5896
Cryogenics(1)	909	909
Suspension	227	227
Shielding(2)	3417	3520
Skids/Landing Gear(2)	2273	2273
Controls/Stability Augmentation	654	654
Power Generation, etc.	4253	4253
Miscellaneous/Contingency	887	887
TOTAL	47,727	51,424

NOTE: (1) Weight driven by number of passengers.
(2) Weight driven by gross weight.

SUMMARY:

- Structural composite weight savings - 7056 lbs (3207 kg) (28% of metallic fuselage structural weight).
- Total vehicle composite weight savings - 8133 lbs (3697 kg)(7% of metallic vehicle total weight).

FIGURE E-5A

SUPPLEMENT D

SECTION F

MAGNEPLANE WEIGHT AND BALANCE

MAGLEV SYSTEM CONCEPT DEFINITION

GENERAL: This appendix covers weight and balance estimates for the Magneplane vehicle. Components have been located so that the vehicle balances near the center. Weight estimates are shown in Figure F-1. The total inertia for the vehicle are shown in Figure F-2 through F-5.

Balance for the large vehicle is more critical and, due to the larger payload of passengers and freight, has a larger variation in center of gravity due to the larger disposable load. The larger vehicle center of gravity limits have been initially established as follows:

Most Forward - Station 56
Most Aft - Station 70

These stations are distanced aft of the nose reference point in feet.

The large vehicle empty center of gravity is at Station 60.8. Empty weight is 66,310 lbs (30,141 kg).

The most forward loading is no freight, and passenger seats full ahead of Station 60.8, and empty aft of this station. The weight and center of gravity for this loading is 80,600 lbs (36,636 kg), cg Station 56.2.

The most aft loading is with the seats ahead of Station 60.8 empty and full seats aft of this station and 8000 lbs (3636 kg) freight in the aft freight compartment; center of gravity for this loading is located at Fuselage Station 69.2. The weight for this loading is 87,980 lb(40,000 kg)s.

This vehicle can be loaded without special considerations for loading as it is essentially impossible to load it outside the limits.

MAGNEPLANE WEIGHT ESTIMATES-lbs
(Composite Airframe)

Item & No.		Small Vehicle Weight Fractions	Small Vehicle Weight (Lbs)	Large Vehicle Weight Fractions	Large Vehicle Weight (Lbs)
(1)	Airframe	.18	9,720	0.17	18,044
(2)	Passengers/Baggage	.18	10,080	.29	30,690
(2)	Freight	-0-	-0-	.076	8,000
(3)	Interior	.044	2,400	.043	4,480
(4)	Heating/Air Cond.	.044	2,400	.034	3,600
(5)	Levitation Magnets	.098	5,364	.054	5,673
(6)	Propulsion Magnets	.083	4,567	.064	6,747
(7)	Cryogenic System	.036	2,000	.019	2,000
(8)	Suspension System	.009	500	.005	500
(9)	Shielding	.064	3,528	.050	5,292
(10)	Skids/Landing Gear	.052	2,880	.048	5,000
(11)	Controls/Stability Augmentation	.017	960	.014	1,440
(12)	Power Pick-up Coils	.058	3,214	.041	4,285
(13)	Battery	.055	3,009	.038	3,970
(14)	Converter	.015	827	.010	1,102
(15)	Miscellaneous	.065	3,551	.040	4,177
TOTAL		1.00	55,000	1.00	105,000

FIGURE F-1

MAGNEPLANE WEIGHT ESTIMATES-kg
(Composite Airframe)

Item & No.		Small Vehicle Weight Fractions	Small Vehicle Weight (kg)	Large Vehicle Weight Fractions	Large Vehicle Weight (kg)
(1)	Airframe	.18	4418	.017	8202
(2)	Passengers/Baggage	.18	4582	.29	13950
(2)	Freight	-0-	-0-	.076	3636
(3)	Interior	.044	1091	.043	2036
(4)	Heating/Air Cond.	.044	1091	.034	1636
(5)	Levitation Magnets	.098	2438	.054	2579
(6)	Propulsion Magnets	.083	2076	.064	3067
(7)	Cryogenic System	.036	909	.019	909
(8)	Suspension System	.009	227	.005	227
(9)	Shielding	.064	1604	.050	2405
(10)	Skids/Landing Gear	.052	1309	.048	2273
(11)	Controls/Stability Augmentation	.017	436	.014	654
(12)	Power Pick-up Coils	.058	1461	.041	1948
(13)	Battery	.055	1368	.038	1804
(14)	Converter	.015	367	.010	501
(15)	Miscellaneous	.065	1614	.040	1899
TOTAL		1.00	25,000	1.00	47,727

FIGURE F1-A

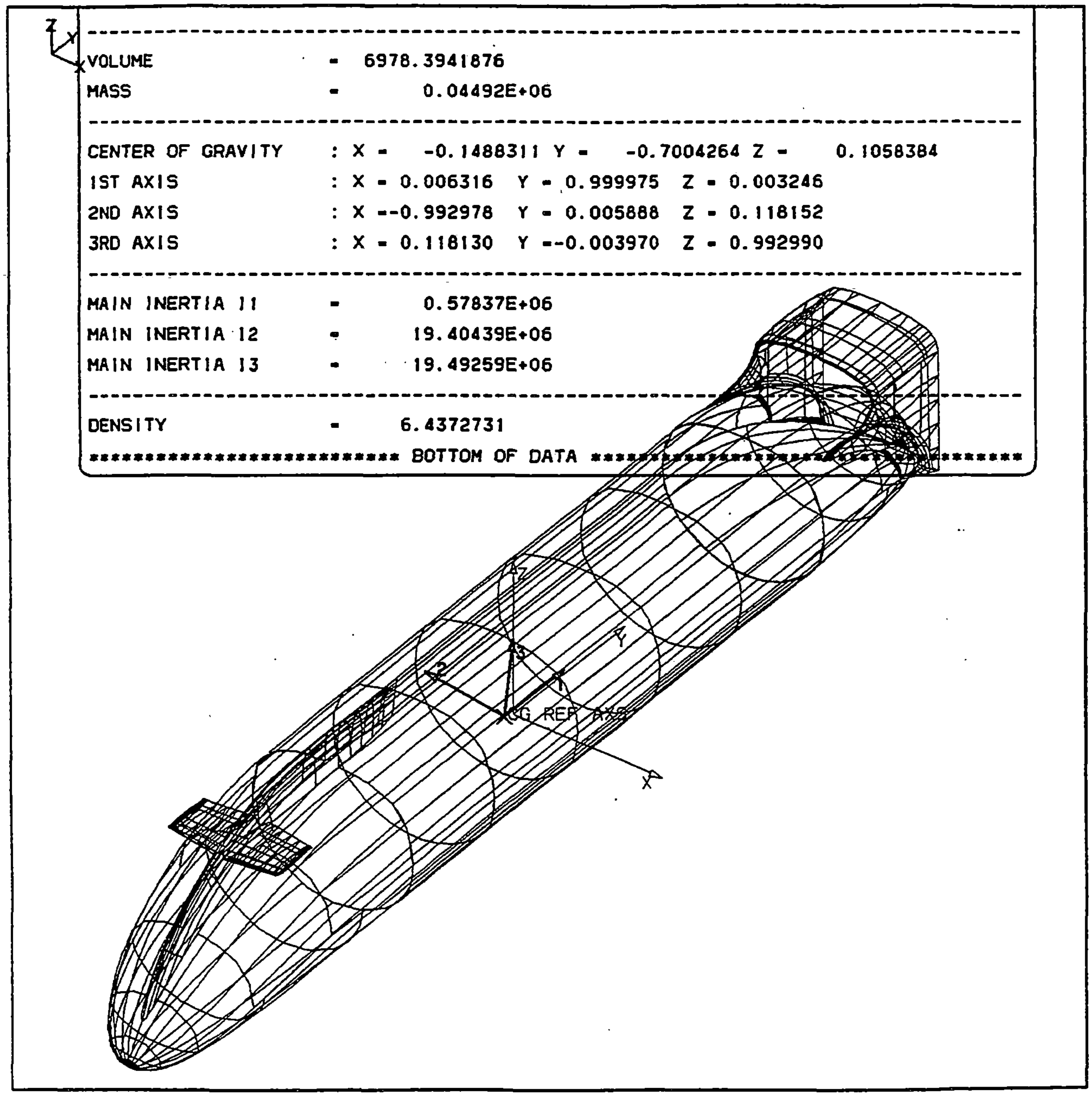


FIGURE F-2 - Inertial Summary for Small Vehicle - Empty

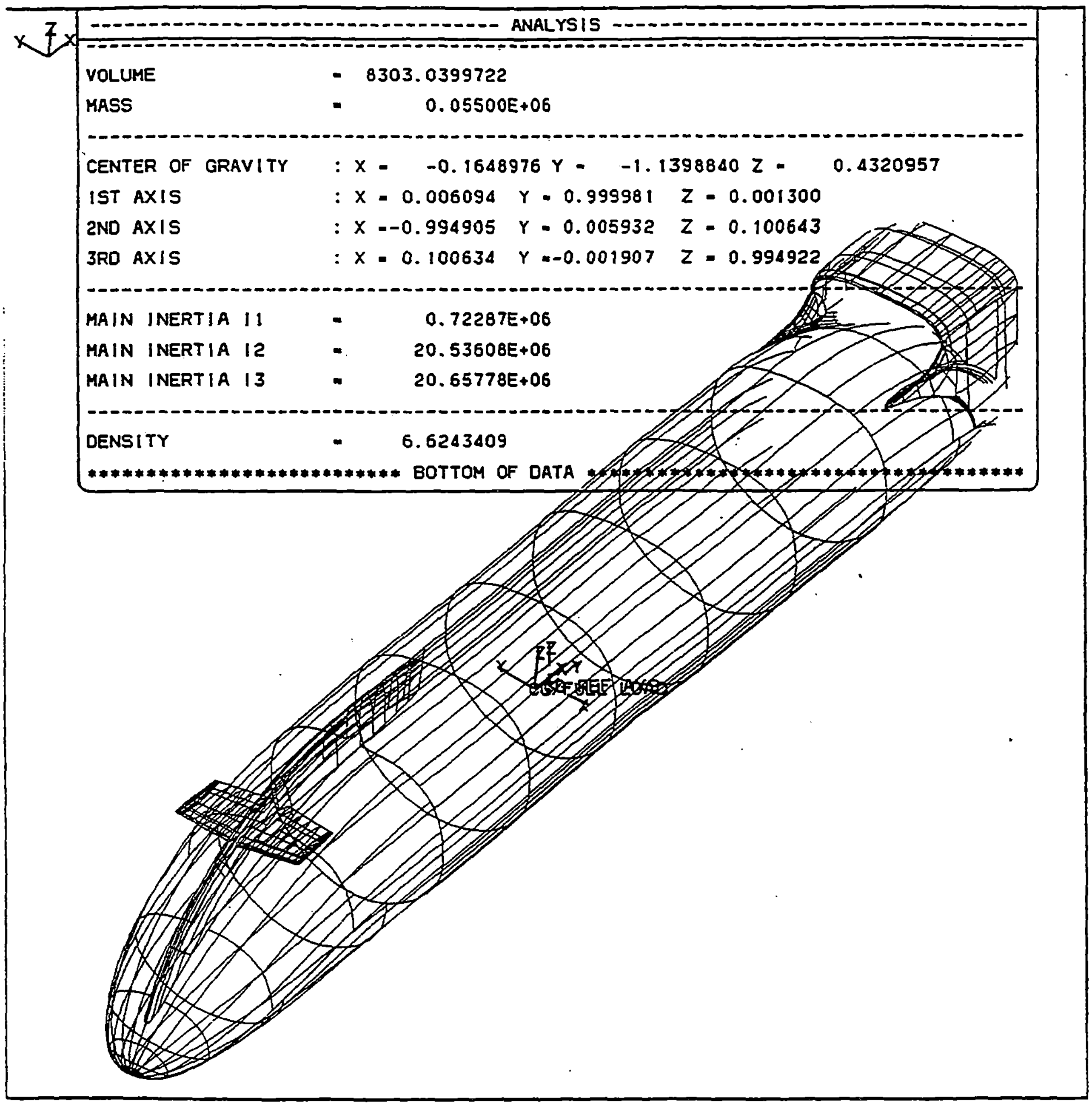


FIGURE F-3 - Inertial Summary for Small Vehicle - Loaded

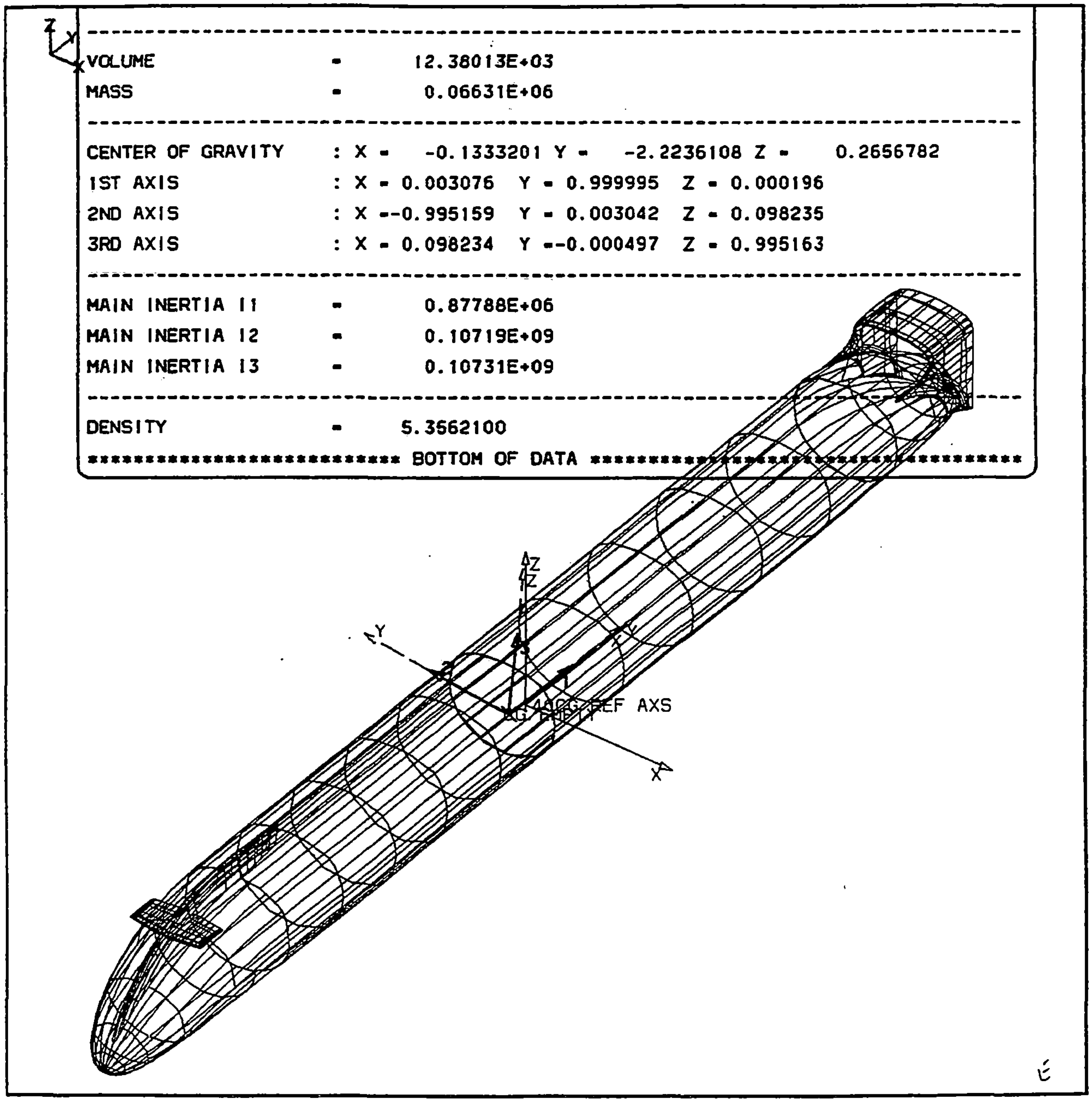


FIGURE F-4 - Inertial Summary for Large Vehicle - Empty
SECTION F

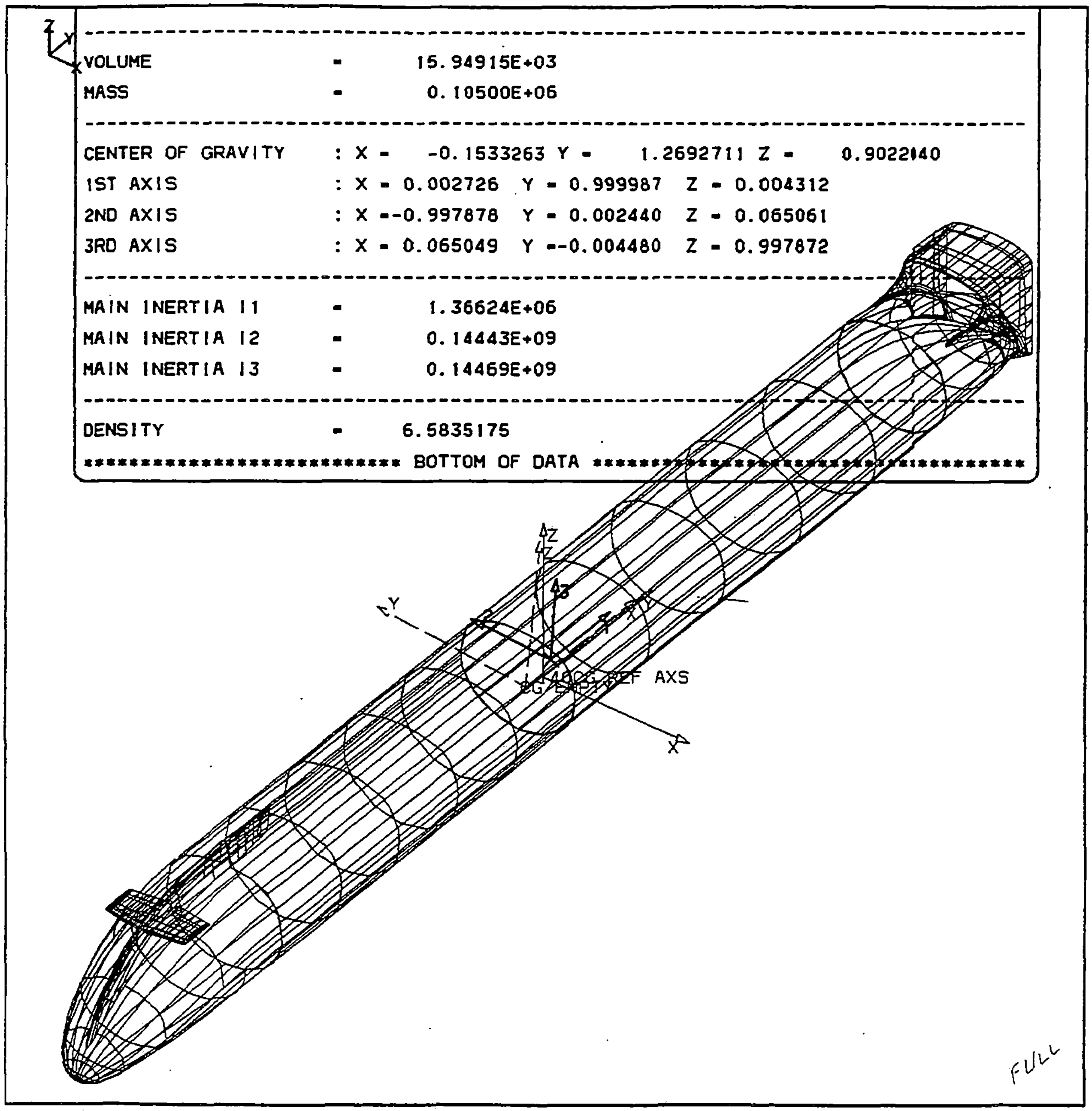


FIGURE F-5 - Inertial Summary for Large Vehicle - Loaded
SECTION F

SUPPLEMENT D
SECTION G

MAGNEPLANE VEHICLE - COST ESTIMATES

1.0 INTRODUCTION: This appendix summarizes engineering cost estimates performed Beech Aircraft Corporation under subcontract to the Massachusetts Institute of Technology for a preliminary design study of a Maglev System Definition. The Maglev vehicle is a magnetically levitated high-speed ground vehicle, which shall be hereafter referred to as the Magneplane. This activity was funded by the Federal Railroad Administration and administered by the U. S. Army Corps of Engineers.

2.0 VEHICLE DESCRIPTION: The Magneplane vehicle is fully described in Appendix A (Model Specification). Cost estimates were made for both the 45 and 140 passenger versions. In addition, cost estimates were made on a metallic variant based on the Materials Trade Study described in Appendix E. The baseline vehicles have composite airframes. The weight estimates for the baseline large (140) passenger) and small (45 passenger) vehicles are shown in Figure G-1A. Weight estimates for the 140 passenger metallic vehicle are shown in Figure G-1B.

SUPPLEMENT D

Figure G-1
MAGNEPLANE WEIGHT ESTIMATES (composite airframe)

Item	Small Vehicle		Large Vehicle	
	weight fraction	weight lbs	weight fraction	weight lbs
(1) Airframe	.18	9,720	.017	18,044
(2) Passengers/Baggage	.18	10,080	.29	30,690
(2) Freight	-0-	-0-	.076	8,000
(3) Interior	.044	2,400	.043	4,480
(4) Heating/Air Cond.	.044	2,400	.034	3,600
(5) Levitation Magnets	.098	5,364	.054	5,673
(6) Propulsion Maganets	.083	4,567	.064	6,747
(7) Cryogenic System	.036	2,000	.019	2,000
(8) Suspension System	.009	500	.005	500
(9) Shielding	.064	3,528	.05	5,292
(10) Skids/Landing Gear	.052	2,880	.048	5,000
(11) Controls/Stability Augmentation	.017	960	.014	1,440
(12) Power Pick-up Coils	.058	3,214	.041	4,285
(13) Battery	055	3,009	.038	3,970
(14) Converter	.015	827	.010	1,102
(15) Miscellaneous	.065	3,551	.040	4,177
TOTAL	**1.00**	**55,000**	**1.00**	**105,000**

SUPPLEMENT D

(metallic airframe)

MAGNEPLANE WEIGHT ESTIMATES FIGURE G-1B

	Items & No.	weight fraction	weight lbs
1)	Airframe	.22	25,100
(2)	Passengers/Baggage	.27	30,690
(2)	Freight	.07	8,000
(3)	Interior	.04	4,480
(4)	Heating/Air Conditining	.03	3,600
(5)	Levitation Magnets	.057	5,925
(6)	Propulsion Magnets	.067	7,047
(7)	Cryogenic System	.02	2,000
(8)	Suspension System	.005	500
(9)	Shielding	.02	5,292
(10)	Skids/landing gear	.05	5,300
(11)	Controls /stability Augmentation	.014	1,440
(12)	Power Pick-up Coils	.01	4,285
(13)	Battery	.04	3,970
(14)	Converter	.005	1,102
(15)	Miscellaneous	.049	4,269
	TOTAL	1.00	113,000

3.0 DEVELOPMENT OF MANUFACTURED WEIGHTS: Weights and cost estimation for flight vehicles have been traditionally based on a manufacured airframe weights, which considered purchased parts separate from manufactured parts. This concept removes from the vehicle, all bolt on purchased parts, so that what remains represents what is "manufactured". The "manufactured" weight includes airframe, wiring, tubing, cables, ducts, etc., but excludes the removed purchased parts. In the case of the Magneplane, some of the system components have been assumed to be partly manufactured and partly purchased. This breakdown is shown in Figure G-2 for both the large and small baseline vehicles, and for the metallic vehicle.

MANUFACTURED WEIGHT RATIONALE

	Composite Airframe				Metallic	
	LARGE VEHICLE		SMALL VEHICLE		LARGE VEHICLE	
Component	Component Wt.(Lbs)	Mfg'd Wt. (Lbs)	Component Wt. (Lbs)	Mfg'd Wt. (Lbs)	Component Wt. (Lbs)	Mfg'd Wt. (Lbs)
1) Airframe	18,044	18,044	9,720	9,720	25,100	25,100
3) Interior	4,480	4,480	2,400	2,400	4,480	4,480
4) Htg & Air Cond.	3,600	1,800	2,400	1,200	3,600	1,800
5&6) Magnets	12,420	----	9,931	-----	12,972	-----
7) Cryogenic System	2,000	100	2,000	100	2,000	100
8) Suspension(Mag Mtg)	500	500	500	500	500	500
9) Shielding	5,292	2,000	2,528	2,000	5,292	2,000
10) Skids/Landing Gear	5,000	5,000	2,880	2,880	5,300	5,300
11) Controls/Stab Aug.	1,440	800	960	700	1,440	800
12) Power P/U Coils	4,285	----	3,214	-----	4,285	-----
13) Battery	3,970	----	3,009	-----	3,970	-----
14) Converter	1,102	220	827	165	1,102	220
15) Miscellaneous/Cont	4,177	4,177	2,031	3,551	4,269	4,269
TOTAL MANUFACTURED WT(LBS)		**37,121**		**23,216**		**44,569**

FIGURE G-2

4.0 DEVELOPMENT OF "MANUFACTURED" WEIGHTS: (Continued) - The following comments on the Manufacturing Weight breakdown relate to the rationale following:

<u>Manufactured Weight Rationale</u>

1)	Airframe	All of the airframe is assumed manufactured.
3)	Interior	All of the interior is assumed manufactured.
4)	Htg & Air Conditioning	Fifty percent is assumed manufactured, the remaining 50% is purchased parts.
5/6)	Magnets	One-hundred percent (100%) of magnets are assumed purchased.
7)	Cryogenic System	Five percent (5%) is assumed manufactured (mounts, lines) and 95% is assumed purchased.
8)	Magnet Suspension	One-hundred percent (100%) is assumed manufactured.
9)	Shielding	Thirty-eight (38%) is assumed manufactured, the rest purchased.
10)	Skids/Landing Gear	One-hundred percent (100%) is assumed manufactured, the rest purchased.
11)	Controls/Stability	Control surfaces, mounts, hinges, Augmentation etc., are assumed to be manufactured. This amounts to 800 lbs for the large vehicle, 700 lbs for the small vehicle. The remainder is assumed purchased.
12)	Pick-up Coils	One hundred percent (100%) is assumed purchased.
13)	Batteries	One hundred percent (100%) is assumed purchased.
14)	Converter	Twenty percent (20%) is assumed manufactured. This would include mounts, wiring, cooling ducts, etc.
15)	Miscellaneous	One hundred percent (100%) is assumed manufactured.

5.0) **DEVELOPMENT COST ESTIMATES:** These estimates were determined by application of an engineering hours per pound against the weight of the particular component. This rationale was applied as follows:

Item 1: Airframe Design was estimated at 100 engineering hours per pound. This is consistent with the C-17 experience which was also composite structure.

Item 3: Interior Design was estimated at 100 engineering hours per pound This was based on an austere commuter interior, some crashworthiness seat design would be required.

Item 4: Heating and Air Conditioning was estimated at 150 engineering hours per pound based on being more complicated than airframe design. The weight of both manufactured and purchased system was used for calculating the design hours.

Item 8: Magnetic mounting (suspension) was estimated at 150 engineering hours per lb.

Item 9: This requirement is to design a magnetic shielding system which will generate an electromagnetic field to cancel the field of the levitation and propulsion magnets in the passenger area. Although the shielding system is estimated to weigh 5,292 lbs for the large vehicle, only 2,000 lbs was assumed to be developed, the remainder would be purchased parts. One-hundred-and-fifty (150) engineering hours per pound were applied for this task.

Item 10: Skids and Landing Gears - Due to the unknowns in the pneumatic skid design, 250 engineering hours per pound were expected. This is an area of new technology with considerable technical risk

Item 11: Stability and Control System Development - This task is to adapt a digital flight computer from a military or other airplane and reprogram it to perform the Magneplane stability augmentation needs. This development includes actuators, control surface design, etc. This was estimated at 250 manhours/lb. This is another high technology area with risk.

Item 12: Power Pick-up Coils, Converter and Electrical Development - Although there are 9,357 lbs of systems in this area, only 220 lbs are considered to require engineering; the remainder being vendor supplied. A total of 150 manhours/lb was estimated. Task would install the pick-up coils, batteries, converters and related electrical components. A separate development for the converter is found in Item 14.

Item 13: Batteries - $100,000 was allocated for battery development by a vendor.

Item 5 and 6: Magnets - $1,000,000 was allocated for magnet development. This number was supplied by MIT.

Item 14: Power Pick-up Coils and Converter Development - $500,000 was allocated for this task. This number was supplied by MIT. This is in addition to the installation design found in Item 1).

- Tooling - 100 manhours/lb. was allocated against the manufactured weight.

The small vehicle used the same factors multiplied by the smaller weights.

- $60 per hour (burdened rate) was used for engineering.
- $50 per hour (burdened rate) for experimental and experimental tooling.

The larger vehicle development cost is summarized in Figure G-3 and the small vehicle in Figure G-4. If both large and small vehicles are concurrently developed, we expect both vehicles can be developed for an additional 30% over large vehicle development costs.

Item 7: Cryogenic System - $500,000 was allocated for this task. This number was agreed on in discussions with Magneplane International This would supply a prototype unit for installation on the prototype unit.

Item 10: Skid Pad Development - $1,000,000 was allocated to develop the skid pads. This number was supplied by MIT.

Systems Integration: A total of 150 engineering hours per pound was allocated against the entire manufactured weight to cover integration of systems and related certification testing.

Test Article Fabrication: The equivalent of nearly three (3) vehicles was assumed for test article fabrication with a twenty (20) manhour/lb experimental fabrication cost allocated.

Materials for Development Articles: Materials were estimated as follows:

- Large Vehicle Prototype
  - 54,132 lbs Composite Materials at $50/lb, 100% Scrap - ($ 5,413K)
  - 57,231 lbs Conventional Materials at $8/lb, 50% Scrap - ($ 687K)
- Small Vehicle Prototype
  - 29,160 lbs Composite Materials at $50/lb, 100% Scrap - ($ 2,916K)
  - 40,488 lbs Conventional Materials at $8/lb, 50% Scrap - ($ 486K)

- **Additional Purchase Parts for Prototype**

Environmental Components	$ 200K
Stability Augmentation Hardware	515K
Power Pick-up Coils	200K
Batteries	38K
Power Converter	300K
Mechanical Hardware	50K
Electrical Hardware	50K
Wiring	50K
Magnets	1,537K
Vehicle Control	450K
Shielding Components	100K
Miscellaneous Components	100K
-Total Prototype Materials -	9,702K
Lge Vehicle	
-Total Prototype Materials -	7,004K
Sml Vehicle	

It was estimated that both large and small vehicle could be developed concurrently for only 30% more cost than the large vehicle developed above.

This would be due to commonalty, combinations of test article, and justification of a large portion of the second vehicle by analysis based on tests of the first vehicle.

The above estimates are based on a composite airframe.

Total vehicle development costs for the large vehicle are shown in Figure G-3 and for the small vehicle in Figure G-4.

Large Magneplane Vehicle Cost Development

Cost Item	Reference Weight	Labor Factor	Manhours hours	Dollars
Airframe Design	18,044 Lbs	100 Mhrs/lb	1,804,400	$ 108,264K
Interior Design	4,480 Lbs	100 Mhrs/lb	448,000	26,880K
Htg & Air Cond Des.	3,600 Lbs	150 Mhrs/lb	540,000	32,400K
Shielding Design	2,000 Lbs	150 Mhrs/lb	300,000	18,000K
Magnet Suspension	500 Lbs	150 Mhrs/lb	75,000	4,500K
Ldg Gear/Skid	5,000 Lbs	250 Mhrs/lb	1,250,000	75,000K
Stability & Control Sys Development	1,440 Lbs	250 Mhrs/lb	360,000	21,600K
Batteries	N/A	-----	-----	100K
Magnet Development	N/A	-----	-----	1,000K
Pwr Converter Dev.	220 Lbs	150 Mhrs/lb	33,000	1,980K
Pick-up Coil Dev.	N/A			500K
Cryogenic Development	2,000 Lbs	-----	-----	500K
Skid Pad Development	-----	-----	-----	1,000K
Miscellaneous	4,177 Lbs	150 Mhrs/lb	626,550	37,593K
Sys. Integration	37,121 Lbs	150 Mhrs/lb	5,568,150	334,089K
Stab. AVG Computer Development	N/A	-----	-----	500K
Test Article Fab.	111,000 Lbs	20 Mhrs/lb	2,220,000	111,000K
Test Article Mtrls.				9,702K
Tooling	37,121	100 Mhrs/lb	3,712,100	185,605K
TOTAL DEVELOPMENT				$ 970,213K

NOTE: Above does not include construction of a test track for testing the vehicle and is based on a composite airframe.

FIGURE G-3

DEVELOPMENT COST ESTIMATE: (Continued)

Small Magneplane Vehicle Development Cost Summary

Cost Item	Reference Weight	Labor Factor	Manhours	Dollars
Airframe Design	9,720 Lbs	100 Mhrs/lb	972,000	$ 58,320K
Interior Design	2,400 Lbs	100 Mhrs/lb	240,000	14,400K
Htg & Air Cond Des.	2,400 Lbs	150 Mhrs/lb	360,000	21,600K
Shielding Design	1,340 Lbs	150 Mhrs/lb	201,000	12,060K
Magnet Suspension	500 Lbs	150 Mhrs/lb	75,000	4,500K
Ldg Gear/Skid	2,880 Lbs	250 Mhrs/lb	720,000	43,200K
Stability & Control Sys Development	960 Lbs	250 Mhrs/lb	240,000	14,400K
Battery Development	N/A	-----	-----	100K
Magnet Development	N/A	-----	-----	1,000K
Pwr Converter Dev.	220 Lbs	150 Mhrs/lb	33,000	1,980K
Pick-up Coil Dev.	N/A	-----	-----	500K
Cryogenic Development	N/A	-----	-----	500K
Skid Pad Development	-----	-----	-----	1,000K
Miscellaneous	3551 Lbs	150 Mhrs/lb	532,650	31,959K
Sys. Integration	23,216 Lbs	150 Mhrs/lb	3,482,400	208,944K
Stab. AVG Computer Development	-----	-----	-----	500K
Test Article Fab.	69,648 Lbs	20 Mhrs/lb	1,392,160	69,648K
Test Article Mtrls:				7,004K
Tooling	23,216	100 Mhrs/lb	2,321,600	$ 116,080K
TOTAL DEVELOPMENT				$ 607,695K

NOTE: Above does not include construction of a test track for testing the vehicle and is based on a composite airframe.

FIGURE G-4

UNIT COST ESTIMATES
Bill of Materials - Production Magneplane
(Composite Vehicle)

ITEM	Large Vehicle Cost-Composite	Small Vehicle Cost-Composite
Composite Materials $50 per lb, 100% Scrap Rate	$ 1,804K	$ 972K
Conventional Materials $8 per lb, 50% Scrap Rate	229K	162K
Cryogenics	88K	88K
Environmental Systems Components	200K	200K
Stability and control System	515K	515K
Power Pick-up Coils	200K	200K
Batteries	50K	50K
Power Converter	300K	300K
Mechanical Hardware	50K	50K
Electrical Hardware	50K	50K
Wiring	50K	50K
Magnets Propulsion	835K	835K
Magnets Levitation	702K	702K
Vehicle Control (Raytheon)	450K	450K
Shielding Components	100K	100K
Miscellaneous Components	100K	100K
TOTAL	**$ 5,723K**	**$ 4,812K**

FIGURE G-5

UNIT COST ESTIMATES
Bill of Materials - Production Magneplane
(Metallic Vehicle)

ITEM	Large Vehicle Cost-Metallic
Metallic Materials $8 per lb, 25% Scrap Rate	$ 535K
Cryogenics	88K
Environmental Systems Components	200K
Stability and Control System	515K
Power Pick-up Couls	200K
Batteries	50K
Power Converer	300K
Mechanical Hardware	50K
Electrical Hardware	50K
Wiring	50K
Magnets - Propulsion	835K
Magnets - Levitation	702K
Vehicle Control	450K
Shielding Components	100K
Miscellaneous Components	100K
TOTAL	**$ 4,225K**

FIGURE G-6

6.0 PRODUCTION COST ESTIMATES: Production cost estimates were made for both the large and small vehicles (composite structure) and for the large vehicle (metallic version). Production costs were based on the bill of materials shown in Figure G-5 and G-6. This was then combined with a manufactured cost estimate. Twenty (20) manhours per lb for Unit 1 was used for composite structure and an 85% learning curve was used for subsequent units. Ten (10) manhours per lb was used for conventional materials and an 85% learning curve. The summary of these estimates are in Figures G-7 and G-8.

SUPPLEMENT D

PRODUCTION COST ESTIMATES - (Composite Vehicle) - FIGURE G-7

Cost Code/ Description	Component	Large vehicle First Article	Average 50 Units	Average 100 Units	Small vehicle :First Article	Average 50 Units	AAvvgv 100 Units
1821 Vehicle Carriage	Airframe Mfg.	21,653	11,041	9,484	11,664	5,947	5,109
1821 Vehicle Carriage	Airframe Pur.	2,033	2,033	2,033	1,134	1,134	1,134
1821 Vehicle Carriage	Shielding Mfg.	1,200	612	526	1,200	612	526
1821 Vehicle Carriage	Shielding Pur.	100	100	100	100	100	100
1821 (Subtotal)		(24,986)	(13,786)	(12,143)	(14,098)	(7,793)	(6,869)
1822 Interiors, etc.	Interior Mfg.	2,688	1,371	1,177	1,440	734	631
Environmental	Htg/Cool/Mfg.	1,080	551	473	720	367	315
Environmental	Htg/Cool/Pur.	200	200	200	200	200	200
1822 (Subtotal)		(3,968)	(2,122)	(1,850)	(2,360)	(1,301)	(1,146)
1823 Levitation & Guid.	Susp.(Mfg).	300	153	131	300	153	131
1823 Levitation & Guid.	Magnet (Pur.	702	702	702	702	702	702
1823 Levitation & Guid.	Cryogenics Mfg.	60	31	26	60	31	26
1823 Levitation & Guid.	Cryogenics Pur.	88	88	88	88	88	88
1823 Levitation & Guid.	Guid. Mfg.	480	245	210	420	214	184
1823 Levitation & Guid.	Guid. Pur	515	515	515	515	515	515
1823 (Subtotal)		(2,145)	(1,734)	(1,672)	(2,085)	(1,703)	(1,646)
1824 Onboard Controls	Raytheon Pur.	450	450	450	450	450	450
1824 (Subtotal)		(450)	(450)	(450)	(450)	(450)	(450)
1825 Propulsion & Brak.	Gr/Skids/Mfg.	3,000	1,530	1, 314	1,728	881	757
1825 Propulsion & Brak.	Magnets Pur.	835	835	835	835	835	835
1825 (Subtotal)		(3,835)	(2,635)	(2 ,149)	(2,536)	(1,716)	(1,592)
1826 Onboard Power	Coils Mfg.	-----	-----	-----	-----	-----	-----
1826 Onboard Power	Coils Pur.	200	200	200	200	200	200
1826 Onboard Power	Converter Mfg.	132	67	59	99	50	43
1826 Onboard Power	Converter Pur.	300	300	300	300	300	300
1826 Onboard Power	Batteries Pur.	50	50	50	38	38	38
1826 Onboard Power	Mech. Pur.	50	50	50	50	50	50
1826 Onboard Power	Elec. Pur.	50	50	50	50	50	50
1826 Onboard Power	Wiring Pur.	50	50	50	50	50	50
1826 (Subtotal)		(832)	(767)	(759)	(787)	(738)	(731)
Miscellaneous Mfg Parts		2,506	1,278	1,098	2,131	1,086	933
Miscellaneous Pur. Parts		100	100	100	100	100	100
Miscellaneous Subtotal		(2,606)	(1,378)	(1,198)	(2,231)	(1,186)	(1,033)
TOTAL		$38,822K	$22,602K	$20,221K	$24,574K	($14,887K	($13,467K

Note: Cost to install systems #1822-1 826 included in #1821. Above costs exclude development.

Figure G-7 Production cost estimates- Composite vehicle

		Large Vehicle		
Cost Code/ Description	Component	First Article	Average 50 Units	Average 100 Units
1821 Vehicle Carriage	Airframe Mf	15,060	7,679	6,596
" " "	Airframe Pr	535	535	535
1821 " " "	Shieldng.Mf	1,200	612	526
" " "	Shieldng.Pr	100	100	100
1821 (Subtotal)		(16,895)	(8,926)	(7,757)
1822 Interior, etc.	Inter. Mfg.	2,688	1,371	1,177
1822 Environmental	Htg/Cool/Mf	1,080	551	473
1822 " " "	" " Pur	200	200	200
1822 (Subtotal)		(3,968)	(2,122)	(1,850)
1823 Levitation & Guid.	Susp.(Mfg)	300	153	131
1823 " " "	Magnet(Pur)	702	702	702
1823 " " "	Cryogenics (Mfg)	60	31	26
1823 " " "	Cryogenics (Pur.)	88	88	88
1823 " " "	Guid. (Mfg)	480	245	210
1823 " " "	Guid.(Pur.)	515	515	515
1823 (Subtotal)		(2,145)	(1,734)	(1,672)
1824 Onbd Controls	RaytheonPur	450	450	450
1824 (Subtotal)		(450)	(450)	(450)
1825 Propulsion & Brak.	Gr/Skids/Mf	3,180	1,621	1,393
1825 " " "	Magnets(Pr)	835	835	835
1825 (Subtotal)		(4,015)	(2,456)	(2,228)
1826 Onboard Power	Coils(Mfg)	—	—	—
1826 " " "	Coils (Pur)	200	200	200
1826 " " "	Convert.Mfg	132	67	59
1826 " " "	Convert.Pur	300	300	300
1826 " " "	Batteries "	50	50	50
1826 " " "	Mech. (Pur)	50	50	50
1826 " " "	Elec. (Pur)	50	50	50
1826 " " "	Wiring(Pur)	50	50	50
1826 (Subtotal)		(832)	(767)	(759)
Miscellaneous Mfg. Parts		2,651	1,306	1,122
Miscellaneous Pur. Parts		100	100	100
Miscellaneous Subtotal		(2,661)	(1,406)	(1,222)
TOTAL		$ 31,055K	$ 17,861K	$ 15,938K
NOTE: Cost to install Sys 1822-26 included in 1821; above excludes dev.				

Figure G-8 Production cost estimates (metalic vehicle)

SUPPLEMENT E: LSM WINDING INDUCTANCE CALCULATIONS

I. Guideway Winding Per-Phase Inductance Calculations

$w := 1.2$	Winding width in meters
$d_b := 2000$	Block length in meters
$p := 0.75$	Pole pitch in meters
$L_{su} := 0.6$	Self inductance per meter in μH (conductors of negligible cross section)
$L_{mu} := 0.86$	Mutual inductance per meter in μH (conductors of negligible cross section 0.01 m apart)

Lsw - Self inductance of one phase

Assumes that adjacent loops of winding are uncoupled.

$$L_{sw} := 2 \cdot d_b \cdot \left[\frac{w}{p} \cdot \left[L_{su} + L_{mu} \right] + L_{su} \right]$$

$L_{sw} = 11744$ μH

Lmw - Mutual inductance between phases

$$L_m(t,d) := 0.2 \cdot t \cdot \left[\ln\left[\frac{t}{d} + \sqrt{1 + \frac{t^2}{d^2}} \right] - \sqrt{1 + \frac{d^2}{t^2}} + \frac{d}{t} \right]$$

Mutual inductance of two conductors of length t and separation d (from Grover, p. 31)

$$L_{ma} := L_m\left[w, \frac{p}{3} \right] \qquad L_{mb} := L_m\left[w, \frac{2 \cdot p}{3} \right]$$

$$L_{mt} := \frac{d_b}{p} \cdot \left[L_{ma} - L_{mb} \right] \cdot 4$$

Mutual inductance of transverse sections

$$L_{me} := 2 \cdot d_b \cdot L_{mu} \cdot \left[\frac{2}{3} - \frac{1}{3} \right]$$

Mutual inductance of end turns

$L_{mw} := L_{mt} + L_{me}$ μH

Lw - Per-phase inductance of the guideway winding

$$L_w := L_{sw} + L_{mw}$$

$L_w = 14212.99$ μH

II. Guideway Winding Per-Phase Resistance

$$d_w := 2 \cdot d_b + \frac{2 \cdot w \cdot d_b}{p}$$

Winding length in meters per phase per conductor

$$d_w = 10400 \quad m$$

$$a_w := \frac{0.02 \cdot p}{6} \cdot 0.60$$

Winding cross-sectional area per phase
h = 0.02 m
fill factor = 60%

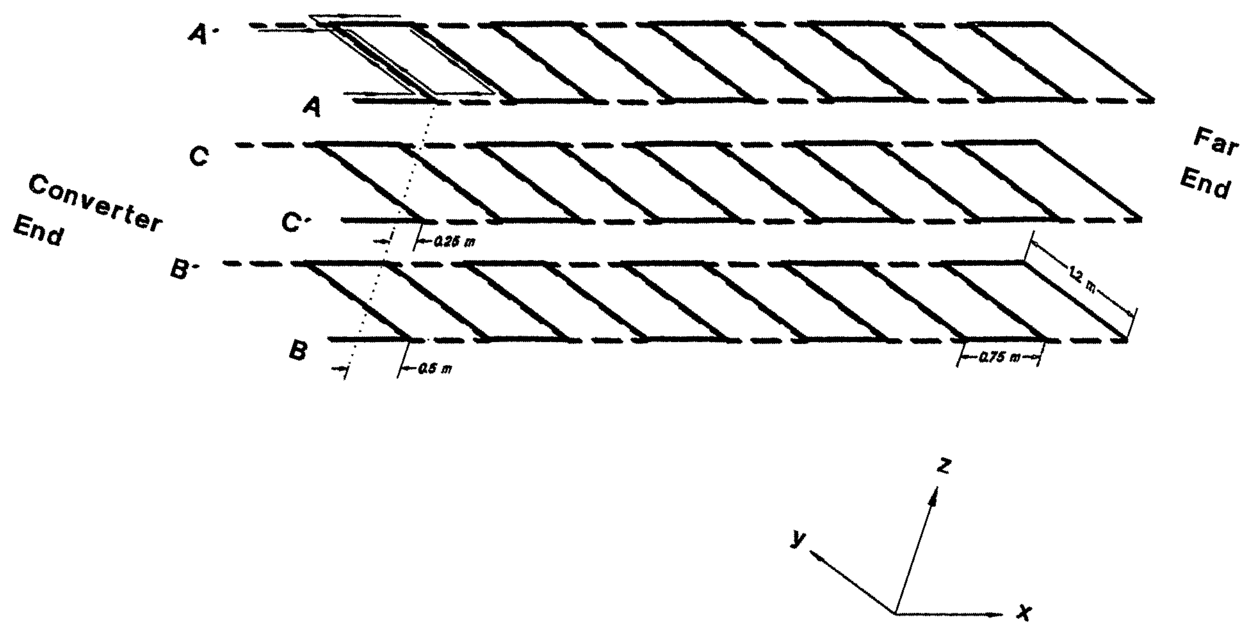
$$\rho := 2.86 \cdot 10^{-8}$$

Resistivity of aluminum in Ω–m

$$R_w := \frac{d_w}{a_w} \cdot \rho$$

$$R_w = 0.2 \quad \Omega$$

Figure 1. Simplified Linear Synchronous Motor Winding (exploded view)



SUPPLEMENT F.

VEHICLE DYNAMIC RESPONSE EQUATIONS

The dynamic response and ride-quality simulation developed and discussed in Section 3.2.2.g.1 of the Final Concept Report is based on the six degree-of-freedom rigid body equations of motion. The formulation of these equations in body fixed axes is presented in this Supplement including all the contributory force and moment terms.

The route structure determines the sequence of grades and horizontal and vertical curves to be followed. The corresponding vehicle speed and bank angle sequences are then obtained from ride comfort considerations and the need to minimize the journey time. The time histories of all the acceleration components are then completely specified throughout the journey. Provided the vehicle enters each of the curves at the correct speed, it will self-bank to perform a coordinated maneuver with the vehicle roll angles equal to the guideway bank angle. The vehicle will be centered on the track, the passengers will not be subject to a lateral acceleration, and the levitation modules will share the normal load equally.

In addition to the prescribed vehicle maneuver dynamics associated with the route structure, there will be a transient dynamic response to each of the maneuvers, modified dynamic response to other disturbances such as track mis-alignments, roughness and flexibility, and gusts and turbulence.

The parameters affecting vehicle dynamic response are:

1) Mass properties.
2) Magnetic and aerodynamic stiffnesses.
3) Magnetic and aerodynamic passive damping.
4) Active control system characteristics.

System excitations and forcing inputs are:

1) Virtual forces resulting from accelerations caused by grades and curves.
2) Guideway mis-alignments, roughness and flexibility.
3) Gusts and turbulence.

The equations governing the vehicle dynamic behavior use body-fixed axes with origin at the center-of-gravity. The axis convention and some of the associated parameters and variables are summarized in Figure F1.

The equations use small perturbation quantities except for bank and roll angles. However, although variables such as pitch and yaw angles will be small, levitation gap changes may be significant compared with the design levitation height, and the non-linear variation of magnetic forces with gap is retained in the equations.

The equations are summarized as follows:

X-force:

$$m\dot{u} = -m\dot{u}_0 + mg(n_F + \theta\cos\phi_0) - D_A(u, u_0, u_G) - D_M(u,u_0,h_n) + T(u,b_n,u_c)$$

where $n_F g$ = fore/aft acceleration for route following.
u_0 = vehicle design speed.

ϕ_0 = design bank angle.
D_A = aerodynamic drag.
D_M = magnetic drag.
u_G = gust or turbulence velocity component.
h_n = gap at the n^{th} levitation module.
T = propulsive thrust.
b_n = gap at the n^{th} propulsion module.
u_c = feedback variable for propulsion force control.

Z-force

$$m(\dot{w} - u_0 q) = -mg(n_V\cos(\phi+\phi_0) + n_H\sin(\phi+\phi_0)) + Z_A(w,u_0,u_G) + Z_{AC}(u_0,\alpha_z) + Z_M(h_n) + Z_{PC}(b_n,w_c)$$

where n_Vg = vertical acceleration for route following.
n_Hg = lateral horizontal acceleration for route following.
Z_A = aerodynamic force component.
Z_{AC} = aerodynamic force control.
α_z = aerodynamic control surface variable.
Z_m = magnetic lift force.
Z_{PC} = magnetic lift force control.
w_c = magnetic control variable.

Pitching-moment:

$$I_{yy}\dot{q} = -I_{yy}\dot{q} + M_A(u,w,u_G,w_G,q) - D_A z_A + M_{AC}(u_0,\alpha_q) - M_M(h_n) + (D_M - T)z_M$$

where M_A = aerodynamic pitching moment.
u_G, w_g = gust velocity components.
z_A = aerodynamic drag moment arm about cg.
M_{AC} = aerodynamic control moment.
α_q = aerodynamic control variable.
M_M = magnetic pitching moment.
z_M = moment arm about cg for the magnetic drag and propulsion forces.

Y-force:

$$m(\dot{v} - u_0 r) = mg(n_H - \cos(\phi+\phi_0) - n_V\sin(\phi+\phi_0)) + Y_A(v,v_G) + Y_{AC}(u_0,\alpha_y) + Y_M(h_n) + Y_{MK}(y_{kn})$$

where Y_A = aerodynamic side-force
v_G = lateral gust velocity components.
Y_{AC} = aerodynamic control side-force.
α_y = aerodynamic control surface variable.
Y_{MK} = magnetic keel side-force.
y_{kn} =lateral movement of vehicle centerline from the guideway centerline at the n^{th} propulsion superconducting magnet location.

Yawing moment:

$$I_{zz}\dot{r} = -I_{zz}\dot{r}_0 + N_A(u_0, r, v_G) + N_{AC}(u_0, \alpha_r) + N_M(h_n) + N_{MK}(y_{kn})$$

where N_A = aerodynamic yawing moment.
N_{AC} = aerodynamic control yawing moment.
α_r = aerodynamic control variable.
N_M = magnetic yawing moment.
N_{MK} = magnetic keel yawing moment.

Rolling moments:

$$I_{xx}\dot{p} = -I_{xx}\dot{p}_0 + L_A(u_0, p, v_G) + L_{AC}(u_0, \alpha_p) + L_M(h_n) + L_{MK}(y_{kn})$$

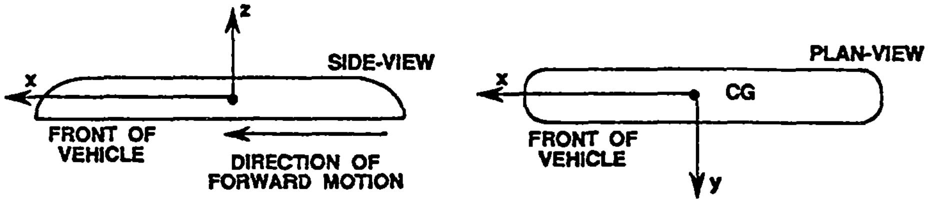
where L_A = aerodynamic rolling moment.
L_{AC} = aerodynamic control rolling moment.
α_p = aerodynamic control variable.
L_M = magnetic rolling moment.
L_{MK} = magnetic keel rolling moment.

The above six equations are supplemented by equations which relate the u,v,w,p,q,r variables to the center-of-gravity location and Euler angles, and by the kinematic relationships between the individual magnetic gaps and the vehicle position and attitude relative to the guideway. The gap relationships also include the effect of guideway mis-alignments and flexibility.

Because of the non-linear behavior of most of the force and moment terms in the equations, they must be solved as time differential equations. A range of suitable algorithms exist for their numerical solution and a simple Heun method was chosen for initial runs. More time-efficient methods may be used later as needed. Initial proving cases include:

1) Response to step changes in guideway alignment including control capability.
2) Transient response in a banked turn.
3) Pitch/heave/fore-aft coupling.

The plunge motion response to a step in the guideway is shown in Figure F2. Feedback control is used with control surface deflection proportional to vertical velocity. Aerodynamic control surfaces are limited in their range before stall occurs and effectiveness is lost. In Figure F2, the feedback is chosen for each step size so that the maximum control deflection is not exceeded. The larger the step size the lower the gain that can be used and the ore resonant the response. At the smallest step size shown, the motion is well-damped from a ride quality viewpoint. More sophisticated feedback control algorithms will make better use of the control power available, for example using non-linear gain. The LQR control design is discussed in Section 3.2.2.g. of the Final Report and ride quality results presented.



BODY - FIXED AXES, ORIGIN AT CENTER-OF-GRAVITY

AXIS	x	y	z
TRANSLATIONAL VELOCITY	u	v (SWAY)	w (PLUNGE)
ROTATIONAL VELOCITY	p (ROLL)	q (PITCH)	r (YAW)
FORCE	X	Y	Z
MOMENT	L	M	N
MOMENT-OF-INERTIA	I_{xx}	I_{yy}	I_{zz}
PRODUCT-OF-INERTIA	I_{yz}	I_{zx}	I_{xy}
ORIENTATION (EULER ANGLES)	ϕ ←	θ ←	ψ

Figure F1. Axis Convention for Vehicle Dynamics.

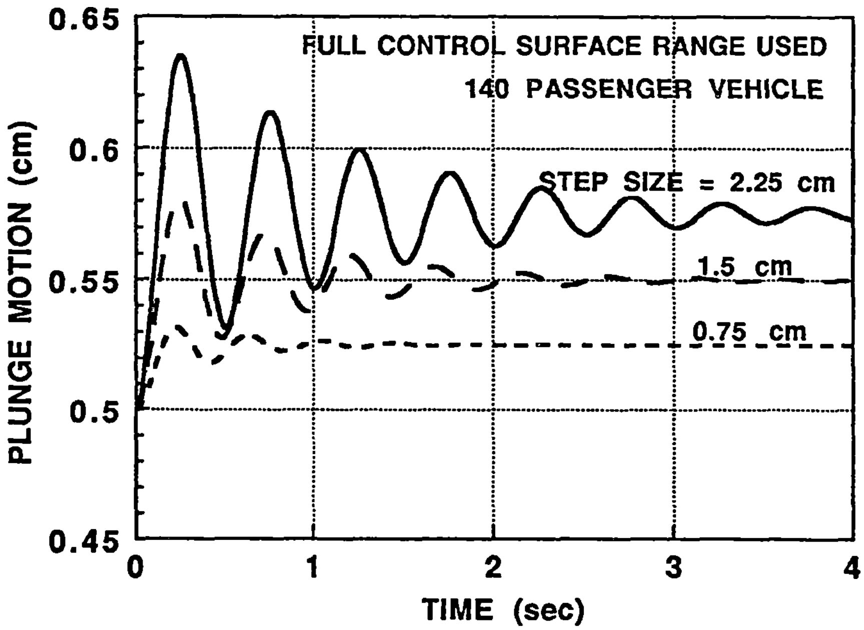


Figure F2. Plunge Response to a Guideway Step.

SUPPLEMENT G: ROUTE ANALYSIS TOOLS

G.1. CONTENTS OF SUPPLEMENT G

The route analysis tools described and listed here are programs written in the C programming language. They were used to accomplish steps 3, 5, and 7 of the route planning procedure as detailed in the Hypothetical Route Report. Other minor programs and spreadsheets were also used in the other steps, but they are not reproduced here.

G.2. NOTES

1. All values are in standard metric units, and units are not noted elsewhere in this supplement. (m, kg, N, s, rad, W, J, etc).

2. File names for .PAR and .RU files (see below) should be limited to four letters, so that the combination can be used for the resulting files. Eg. P1.PAR + HR12.RU => P1HR12.RUP.

G.3. DESCRIPTION OF THE FILE TYPES

RU FILES

ASCII Route description entered by user. The first line contains "S" followed by the distance measurement in meters of both endpoints of the route. The following lines list horizontal and vertical curves. The two types of curves can either be interspersed or one can follow the other. In either case the distance measurements must be in order. For vertical curves begin the line with "V" and supply the position, length, the grade preceding the curve and the grade following the curve, (in decimal, not percent). For horizontal curves, begin the line with "H" and supply the position, radius, and degrees of curvature. Example:

```
s    0     2000
h    800   400   20
v    1020  900   .08   .04
h    1800  800   60
```

RUP FILES

Processed route description. This is a binary file, not readable except through the program RUPREP. It is a table of segments in the route and contains the following fields for each segment:

position of seg
grade
curvature (1 / radius)
velocity window 1 (0 to some max)
velocity window 2 (0 or some min to some (possibly greater) max)
calculated attainable velocity
time and cumulative time
power, energy, and cumulative energy
forward acceleration (last seg to this seg)
vertical and lateral acceleration
roll angle
optimal winding type for this seg (normal or low-R)

Refer to the code for the exact fields.

PAR FILES

ASCII parameter file. Contains the values for the parameters below. There are two values for each of the vehicle fields (for small and large vehicles respectively) and two values for each of the cost fields (initial and per-second respectively). Ride quality and system fields contain one value. When two values are expected, separate them with one or more spaces. Otherwise separate parameters by newlines.

A spreadsheet containing the four PAR files actually used is printed immediately following this text.

SEG FILES

ASCII version of RUP file.

BLO FILES

ASCII block file. Contains values about optimized costs, and a listing of blocks, each with power requirements and block type.

CUR FILES

ASCII curve file. Contains information about each horizontal curve.

G.4. DESCRIPTION OF THE PROGRAMS

MAKERUP: Reads specified RU file and builds a RUP file with coordinated curves, grades, and some curve velocity limits.

VCALC: Processes a .RUP file and fills in the fields for instantaneous velocity based on the curve data in each segment.

VCALCT: Processes a .RUP file and fills in the fields time, acceleration, power, energy, and real maximum velocity.

RUPREP: Converts a .RUP file to a readable SEG file.

COSTOPT: Takes a .RUP file, finds minimum costs, and produces a .BLO file.

REVERSE: Reverses a RU file given a RU file, so that the output is the route in the reverse direction.

G.5. DOS BATCH FILE TO PROCESS ROUTE

The following batch file expects to see a PAR file and a RU file, and it creates all the other file types accordingly. The command line arguments are:

1. "GO"
2. <subdirectory> (where it will find all files except parameter files)
3. <ru-file-name>
4. <par-file-name>
5. <resolution> (meters per segment of analysis)

```
@rem GO.BAT
@echo off
echo USAGE: go subdir ru-file param-file resolution
echo -----
echo Parameter file:        params\%3.par
echo Route file:            %1\%2.ru
echo Outputs are...
echo    Processed route:     %1\%2%3.rup
echo    Curve report:        %1\%2%3.cur
echo    Route report:        %1\%2%3.seg
echo    Blocks/optimization: %1\%2%3.blo
pause
```

```
makerup params\%3.par %1\%2.ru %1\%2%3.rup %4 %1\%2%3.cur
vcalc params\%3.par %1\%2%3.rup
vcalct params\%3.par %1\%2%3.rup
ruprep %1\%2%3.rup %1\%2%3.seg 3
pause
costopt params\%3.par %1\%2%3.rup %1\%2%3.blo
```

G.6. PARAMETER SPREADSHEET

See following page.

G.7. CODE

See following pages.

SYSTEM PARAMETERS

		No limits NONE	
RIDE QUALITY	system max velocity	134	
	forward min acceleration	-999	
	forward max acceleration	999	
	emergency deceleration (positive)	999	
	up min acceleration	-999	
	up max acceleration	999	
	max roll angle (deg)	89	
	max roll rate (deg)	999	
		0	
	max vertical jerk	999	
	max long jerk	999	
	max total vector acceleration	999	
		0	
VEHICLE (small, large)	capacity	45	140
	mass	25000	50000
	standing normal weight	302528	605055
	aero drag constant	0.85	1.0704
	on-board power incl efficiency	0	0
	drag mode 2 (*v)	322	593
	drag mode 2 (c)	6231	13426
	lift/drag constant 1	-0.12686	-0.12686
	lift/drag constant 2	0.31863	0.31863
	lift/drag constant 3	-0.00033	-0.00033
	skid coefficient of friction	0.05	0.05
SYSTEM	passengers per second required	2.66666	
	time of optimization	1.2E+09	
	frac pas served by small vehicles	0	
	mech power limit	1.0E+35	
	utility-guideway power factor	1	
	guideway-utility power factor	1	
	winding resistance per meter	0	
	min velocity for mode 2	30	
	min velocity for mode 3	50	
	thrust threshhold for low-R LSM	110000	
	winding I-F ratio	0.0215	
COSTS (initial, per sec)	per meter w/o lift - fixed	7188	0
	per meter w/lift - fixed	7188	0
	delta per meter for low-R LSM	300	0
	per small vehicle	2500000	0
	per large vehicle	5000000	0
	per block - fixed	3000000	0
	per block - per Watt	0.1	0
	per switch	10000000	0.03171
	per Watt	0	2.4E-08

Design Goal BEST		Minimum Requirement MIN-S		Seated/Belted MIN-B	
134		134		134	
-1.48		-1.82		-3.6	
1.48		1.82		3.6	
4.9		4.9		4.9	
9.31		8.82		8.82	
11.76		12.74		13	
24		30		45	
5		5		10	
0		0		0	
0.98		2.94		2.94	
0.686		2.45		2.45	
2.352		3.528		3.92	
0		0		0	
45	140	45	140	45	140
25000	50000	25000	50000	25000	50000
302528	605055	302528	605055	302528	605055
0.85	1.0704	0.85	1.0704	0.85	1.0704
250000	250000	250000	250000	250000	250000
322	593	322	593	322	593
6231	13426	6231	13426	6231	13426
-0.12686	-0.12686	-0.12686	-0.12686	-0.12686	-0.12686
0.31863	0.31863	0.31863	0.31863	0.31863	0.31863
-0.00033	-0.00033	-0.00033	-0.00033	-0.00033	-0.00033
0.05	0.05	0.05	0.05	0.05	0.05
2.66666		2.66666		2.66666	
1.2E+09		1.2E+09		1.2E+09	
0		0		0	
15300000		20400000		20400000	
1.071		1.071		1.071	
0.931		0.931		0.931	
0.0001		0.0001		0.0001	
30		30		30	
50		50		50	
110000		110000		110000	
0.0215		0.0215		0.0215	
7188	0	7188	0	7188	0
7188	0	7188	0	7188	0
300	0	300	0	300	0
2500000	0	2500000	0	2500000	0
5000000	0	5000000	0	5000000	0
3000000	0	3000000	0	3000000	0
0.1	0	0.1	0	0.1	0
10000000	0.03171	10000000	0.03171	10000000	0.03171
0	2.4E-08	0	2.4E-08	0	2.4E-08

```
#include "stdio.h"
#include "stdlib.h"
#include "math.h"
#include "ctype.h"
#include "conio.h"
#include "fcntl.h"
#include "sys\stat.h"
#include "io.h"

#define g 9.8                        /*acceleration of gravity*/
#define g2 g * g
#define pi 3.141592654
```

```
/*code to read parameter file: include in any tool*/

#define PARBUFLEN 80

/*ride quality.  1=long. 2=vert. 3=lateral  h=roll*/
float vmax;
float amax1, amin1;
float decel;
float amax2, amin2;
float hmax, hamax;
float jmax2, jmax1;
float amax123;

/*vehicle*/
int vehcapacity[2];
float vehmass[2];
float vehweight[2];
float AE1[2];
float Photel[2];
float drag2a[2], drag2c[2];
float LD1[2];
float LD2[2];
float LD3[2];
float skidfriction[2];

/*system*/
float pps;
float opttime;
float fracveh0;
float maxpower;
float peff, peffback, Rperm;
float vmode2, vmode3, thrustthresh, ItoF;

/*costs ([0] is initial, [1] is per second thereafter*/
float costg[2];
float costgl[2];
float costlowR[2];
float costv0[2];
float costv1[2];
float costb[2];
float costbW[2];
float costsw[2];
float costW[2];

void readone(FILE *fp, char *buf, float *a)
  {
  fgets(buf, PARBUFLEN - 1, fp);
  sscanf(buf, "%f", a);
  }

void readtwo(FILE *fp, char *buf, float *a)
  {
  fgets(buf, PARBUFLEN - 1, fp);
  sscanf(buf, "%f%f", a, a + 1);
  }

void readpar(char *name)
  {
  float temp[2], dum;
  FILE *fp;
  char buf[PARBUFLEN];

  if ((fp = fopen(name, "rt")) == NULL) {
```

```
		puts("Can't read parameter file.");
		exit(1);
		}
	/*ride quality*/
	readone(fp, buf, &vmax);
	readone(fp, buf, &amin1);
	readone(fp, buf, &amax1);
	readone(fp, buf, &decel);
	readone(fp, buf, &amin2);
	readone(fp, buf, &amax2);
	readone(fp, buf, &hmax);
	hmax = hmax / 180 * 3.14159;   /*input is degrees, cvt to rad*/
	readone(fp, buf, &hamax);
	hamax = hamax / 180 * 3.14159;   /*input is degrees, cvt to rad*/
	readone(fp, buf, &dum);
	readone(fp, buf, &jmax2);
	readone(fp, buf, &jmax1);
	readone(fp, buf, &amax123);
	readone(fp, buf, &dum);

	/*vehicle*/
	readtwo(fp, buf, temp);
	vehcapacity[0] = temp[0];
	vehcapacity[1] = temp[1];
	readtwo(fp, buf, vehmass);
	readtwo(fp, buf, vehweight);
	readtwo(fp, buf, AE1);
	readtwo(fp, buf, Photel);
	readtwo(fp, buf, drag2a);
	readtwo(fp, buf, drag2c);
	readtwo(fp, buf, LD1);
	readtwo(fp, buf, LD2);
	readtwo(fp, buf, LD3);
	readtwo(fp, buf, skidfriction);

	/*system*/
	readone(fp, buf, &pps);
	readone(fp, buf, &opttime);
	readone(fp, buf, &fracveh0);
	readone(fp, buf, &maxpower);
	readone(fp, buf, &peff);
	readone(fp, buf, &peffback);
	readone(fp, buf, &Rperm);
	readone(fp, buf, &vmode2);
	readone(fp, buf, &vmode3);
	readone(fp, buf, &thrustthresh);
	readone(fp, buf, &ItoF);

	/*costs*/
	readtwo(fp, buf, costg);
	readtwo(fp, buf, costgl);
	readtwo(fp, buf, costlowR);
	readtwo(fp, buf, costv0);
	readtwo(fp, buf, costv1);
	readtwo(fp, buf, costb);
	readtwo(fp, buf, costbW);
	readtwo(fp, buf, costsw);
	readtwo(fp, buf, costW);

	fclose(fp);
	puts("Read parameter file.");
	}
```

```
/*code to deal with rup files*/

/*each segment in layout*/
struct rupseg {
  long start, end;        /*distance at start and end of seg - set by makerup*/
  float d;                /*grade in seg - makerup*/
  float ir;               /*1/radius of curvature in seg - makerup*/
  float bank;             /*bank angle of guideway in seg - vcalct*/
  float cost[2];          /*cost of guideway construction - costopt*/
  float vmax1, vmin2, vmax2;
                          /*two ranges of allowable velocity. if
                          vmin2==0 then there is no second range
                          and vmax1 = vmax2
                          vmax2 is always the absolute max - vcalc/vcalct*/
  float vmax;             /*real velocity limit with ride q included - makerup/vcalc/vcalct*/
  float time;             /*travel time at vmax through this seg - vcalct*/
  float cumtime;          /*cumulative travel time up to end of seg - vcalct*/
  float drag;             /*sum of all drag types - vcalct*/
  int promode;            /*propulsion mode 1,2,3 : if 1 then there
                          is no lev plate. 2/3 differentiate between
                          use of skids or not at velocity vmax - vcalct*/
  float w;                /*power used in segment - vcalct*/
  float E;                /*energy used in segment - vcalct*/
  float cumE;             /*cumulative energy used up to end of seg - vcalct*/
  float a1, a2, a3;       /*actual fwd/vert/lat acceleration - vcalct*/
  float roll;             /*actual roll angle - vcalct*/
  };

/*variables that are always accurate after rupopen*/
long rup_nrec;                    /*number of segments in current file
                          Note that seg 0 and n are not real*/
int rupfile;
long rupbufsiz;

/*variables that are only accurate after set by rupset*/
float smax;                       /*total distance covered*/
float rup_s0, rup_send;           /*beginning and end distances*/
float tmax;                       /*total time*/
float Emax;                       /*total energy*/


void rupopen(char *name, int flags)
  {
  if ((rupfile = open(name, O_RDWR|O_BINARY|flags, S_IREAD|S_IWRITE)) < 0) {
        puts("Can't open route file (rup).");
        exit(1);
        }
  lseek(rupfile, 0, 2);
  rupbufsiz = sizeof(struct rupseg);
  rup_nrec = tell(rupfile) / rupbufsiz;
  }

void rupread(struct rupseg *s, long rec)
  {
  if (lseek(rupfile, rec * rupbufsiz, 0) == -1
   || read(rupfile, s, rupbufsiz) != rupbufsiz) {
        perror("Read from rup file");
        exit(31);
        }
   }

void rupwrite(struct rupseg *s, long rec)
  {
```

```
    if (lseek(rupfile, rec * rupbufsiz, 0) == -1
     || write(rupfile, s, rupbufsiz) != rupbufsiz) {
            perror("Write to rup file");
            exit(31);
            }
     if (rec >= rup_nrec) ++rup_nrec;
     }

void rupclear(struct rupseg *s)
    {
    s->d = s->ir = s->vmax1 = s->vmin2 = s->vmax2
    = s->vmax = s->time = s->cumtime = s->w = s->E
    = s->cumE = s->a1 = s->roll = 0;
    s->a2 = g;
    s->bank = s->drag = s->cost[0] = s->cost[1] = 0;
    s->promode = 0;
    s->start = s->end = 0;
    }


void rupset()
    /*RUP-SET, not R-UPSET !*/
    {
    struct rupseg s;
    if (rup_nrec < 2) {
            smax = tmax = Emax = 0;
            return;
            }

    rupread(&s, rup_nrec - 2);
    rup_send = s.end;
    tmax = s.cumtime;
    Emax = s.cumE;
    rupread(&s, 0);
    rup_s0 = s.end;
    smax = rup_send - rup_s0;
    }
```

```
/*code to load a ru file: include in any tool*/

#define RU_MAXCURV 200

float ru_smin, ru_smax;                 /*endpoints of route*/
int ru_nhcurv, ru_nvcurv;              /*number of curves*/

struct {
  float s, r;     /*point, radius*/
  float curve;    /*radians*/
  } ru_hcurv[RU_MAXCURV];        /*list of horiz curves*/

struct {
  float s, L;     /*point, curve length*/
  float d1, d2;   /*grade before and after*/
  } ru_vcurv[RU_MAXCURV];        /*;ost of vert curves*/


void readru(char *name)
  /*reads in given file into above variables*/
  {
  FILE *fi;
  char buf[80], *buf2;
  int i;

  if ((fi = fopen(name, "rt")) == NULL) {
          puts("Can't read input file.");
          exit(3);
          }

  ru_nhcurv = ru_nvcurv = 0;
  while (fgets(buf, 80, fi) != NULL) {
          buf2 = buf + 1;
          switch (toupper(buf[0])) {
                  case 'S':
                  if (ru_smax) {
                          puts("Total distance reported twice.");
                          exit(6);
                          }
                  sscanf(buf + 1, "%f%f", &ru_smin, &ru_smax);
                  if (ru_smin < 0 || ru_smax <= ru_smin) {
                          puts("Total distance is negative or zero.");
                          exit(5);
                          }
                  break;

                  case 'H':
                  if (ru_nhcurv >= RU_MAXCURV) {
                          puts("Too many horizontal curves.");
                          exit(10);
                          }
                  sscanf(buf2, "%f%f%f", &ru_hcurv[ru_nhcurv].s,
                   &ru_hcurv[ru_nhcurv].r, &ru_hcurv[ru_nhcurv].curve);
                  if (ru_hcurv[ru_nhcurv].s < 0
                   || ru_hcurv[ru_nhcurv].r <= 0
                   || ru_hcurv[ru_nhcurv].curve <= 0
                   || ru_hcurv[ru_nhcurv].curve >= 360) {
                          printf("Invalid horizontal curve (s = %f).\n",
                           ru_hcurv[ru_nhcurv].s);
                          exit(7);
                          }
                  ru_hcurv[ru_nhcurv].curve *= 3.14159 / 180;
                  ++ru_nhcurv;
```

```
                        break;

                        case 'V':
                        if (ru_nvcurv >= RU_MAXCURV) {
                                puts("Too many vertical curves.");
                                exit(10);
                                }
                        sscanf(buf + 1, " %f %f %f %f", &ru_vcurv[ru_nvcurv].s,
                         &ru_vcurv[ru_nvcurv].L, &ru_vcurv[ru_nvcurv].d1,
                         &ru_vcurv[ru_nvcurv].d2);
                        if (ru_vcurv[ru_nvcurv].s < 0
                         || ru_vcurv[ru_nvcurv].L <= 0) {
                                printf("Invalid vertical curve (s = %f).\n",
                                 ru_vcurv[ru_nvcurv].s);
                                exit(8);
                                }
                        ++ru_nvcurv;
                        break;

                        otherwise:
                        puts("Unrecognized letter in column 0.");
                        exit(9);
                        }
                }

        /*check for order*/
        for (i = 1; i < ru_nhcurv; ++i)
                if (ru_hcurv[i].s < ru_hcurv[i - 1].s) {
                        printf("Horizontal curve out of order (s = %f).\n",
                         ru_hcurv[ru_nhcurv].s);
                        exit(12);
                        }
        for (i = 1; i < ru_nvcurv; ++i)
                if (ru_vcurv[i].s < ru_vcurv[i - 1].s) {
                        printf("Vertical curve out of order (s = %f).\n",
                         ru_vcurv[ru_nvcurv].s);
                        exit(12);
                        }

        /*check for corresponding grades*/
        for (i = 1; i < ru_nvcurv; ++i)
                if (ru_vcurv[i].d1 != ru_vcurv[i - 1].d2) {
                        printf("Adjacent grade values don't match (s = %f).\n",
                         ru_vcurv[ru_nvcurv].s);
                        exit(13);
                        }

        puts("Read route file (.ru)");
        }
```

```
/*converts a ru file to a rup file and dumps curve information
parameters are input-file output-file segment-length report-file
*/

#include "route.h"
#include "ru.cpp"
#include "rup.cpp"
#include "par.cpp"

/*structures to hold interim H and V curv data*/
struct {
  char seen;                          /*output yet?*/
  float L, dd;                        /*info for sine*/
  float Ls, Lc, v;                    /*info for tapered circle*/
  } hc[RU_MAXCURV];

struct {
  char seen;                          /*output yet?*/
  float v;                            /*velocity limit for jerk*/
  } vc[RU_MAXCURV];

float dsin(long s, long L, long ctr, float d1, float d2)
  /*returns the slope at s using a sin function where
  L is half the wavelength, or the length of the curve
  ctr is the center point of the curve
  d1 is the slope at ctr - L/2
  d2 is the slope at ctr + L/2
  */
  {
  return (d2 + d1) * 0.5 + sin(pi * (s - ctr) / L) * (d2 - d1) * 0.5;
  }

void main(int argc, char *argv[])
  {
  struct rupseg seg, prevseg;
  long i, j, k, seglength, midpt, halfL, L, j0, j1, k0, k1, k2, k3;
  float slope0, slope1, vmost;
  float vlimvert, vlimhorz;           /*velocity limits vert/horz*/
  FILE *fo;

  /*evaluate parameters*/
  if (argc != 6) {
          puts("Usage: MAKERUP <param-file> <in-file> <out-file> <seg-length> <report-file>");
          exit(1);
          }
  if ((seglength = atoi(argv[4])) <= 0) {
          puts("Segment length must be nonzero.");
          exit(2);
          }

  /*get parameter file*/
  readpar(argv[1]);

  /*get input file*/
  readru(argv[2]);

  /*mark all vert curves as not output yet*/
  for (i = 0; i < ru_nvcurv; ++i)
          vc[i].seen = 'N';

  /*process horizontal curves*/
  for (i = 0; i < ru_nhcurv; ++i) {
          /*find vmax in this curve for roll angle*/
```

```
		hc[i].v = sqrt(tan(hmax) * g * ru_hcurv[i].r);

		/*find vmax in this curve for roll rate*/
		vmost = g * hamax * ru_hcurv[i].r * ru_hcurv[i].r
		 * ru_hcurv[i].curve;
		vmost = pow(vmost, (float)1/3);
		if (hc[i].v > vmost)
			hc[i].v = vmost;

		/*set lengths of spiral and circle segments of this curve*/
		hc[i].seen = 'N';
		hc[i].Ls = pow(hc[i].v, 3) / (g * hamax * ru_hcurv[i].r);
		hc[i].Lc = ru_hcurv[i].r * ru_hcurv[i].curve - hc[i].Ls;

		/*for sine curves , not used now*/
		hc[i].L = pi * ru_hcurv[i].r / tan((pi - ru_hcurv[i].curve) / 2);
		hc[i].dd = -2 * tan(ru_hcurv[i].curve / 2);
		}

	/*process vertical curves*/
	for (i = 0; i < ru_nvcurv; ++i) {
		vc[i].v = 2 * jmax2 * ru_vcurv[i].L * ru_vcurv[i].L
		 / (ru_vcurv[i].d2 - ru_vcurv[i].d1) / pi / pi;
		if (vc[i].v > 0) /*cube root of + or - number*/
			vc[i].v = pow(vc[i].v, (float)1/3);
		else
			vc[i].v = -pow(-vc[i].v, (float)1/3);
		}

	/*write all records in rup file.
	Note that there are two extra segments of length zero at the
	beginning and end of the file.*/
	rupopen(argv[3], O_CREAT|O_TRUNC);
	rupclear(&seg);
	rupclear(&prevseg);
	prevseg.end = ru_smin;
	j = 0;				/*vert curve number*/
	k = 0;				/*horiz curve number*/
	slope0 = ru_nhcurv ? hc[0].dd / -2 : 0;
					/*incomming horiz "slope"*/

	/*for each record */
	for (i = ru_smin; i <= ru_smax; i += seglength) {
		seg.start = prevseg.end;
		seg.end = i;
		midpt = (i == ru_smin) ? i : i - seglength / 2;
		vlimvert = vlimhorz = vmax;

		/*figure out which vertical curve is here if any
		and set grade and vlimvert*/
		testwithinj:
		halfL = ru_vcurv[j].L / 2;
		j0 = ru_vcurv[j].s - halfL;
		j1 = ru_vcurv[j].s + halfL;
		if (!ru_nvcurv)
			seg.d = 0;
		else if (midpt >= j0 && midpt <= j1) {
			seg.d = dsin(midpt, ru_vcurv[j].L, ru_vcurv[j].s,
			 ru_vcurv[j].d1, ru_vcurv[j].d2);
			vc[j].seen = 'Y';
			vlimvert = vc[j].v;
			}
		else if (midpt < j0)
```

```
                        seg.d = ru_vcurv[j].d1;
                else {                          /*after curve j*/
                        if (j >= ru_nvcurv - 1)
                                seg.d = ru_vcurv[j].d2;
                        else {
                                ++j;
                                goto testwithinj;
                                }
                        }

        /* SINE CURVES:
                /*figure out which horizontal curve is here if any
                and set slope. If there is none, set slope1 = slope0
                If there is a curve here,
                set slope1 = slope here. First remember previous
                slope.*/

                testwithink:
                j0 = ru_hcurv[k].s - hc[k].L / 2;
                j1 = ru_hcurv[k].s + hc[k].L / 2;
                if (!ru_nhcurv)
                        slope1 = slope0;
                else if (midpt >= j0 && midpt <= j1) {
                        slope1 = dsin(midpt, hc[k].L, ru_hcurv[k].s,
                         hc[k].dd / -2, hc[k].dd / 2);
                        hc[k].seen = 'I'; /*now within curve*/
                        }
                else if (midpt < j0)            /*before curve k*/
                        slope0 = slope1 = hc[k].dd / -2;
                else {                          /*after curve k*/
                        if (hc[k].seen == 'I') {
                                slope1 = hc[k].dd / 2;
                                hc[k].seen = 'Y';
                                }
                        else if (k < ru_nhcurv - 1) {
                                ++k;
                                goto testwithink;
                                }
                        }

                /*set the curvature based on the horiz slope difference*/
                seg.ir = (atan(slope0) - atan(slope1)) / seglength;
                slope0 = slope1;

        TAPERED CIRCLES: */
                /*figure out which horizontal curve is here if any
                and set curvature and vlimhorz.*/
                testwithink:
                k0 = ru_hcurv[k].s - hc[k].Lc / 2 - hc[k].Ls;
                k1 = ru_hcurv[k].s - hc[k].Lc / 2;
                k2 = ru_hcurv[k].s + hc[k].Lc / 2;  /*can be < k1!!*/
                k3 = ru_hcurv[k].s + hc[k].Lc / 2 + hc[k].Ls;
                if (!ru_nhcurv) {
                        seg.ir = 0;
                        }
                else if (midpt < k0) {          /*before curve k*/
                        seg.ir = 0;
                        }
                else if (midpt >= k0 && midpt <= k1
                 && midpt <= ru_hcurv[k].s) {   /*in spiral*/
                        seg.ir = 1 / ru_hcurv[k].r * (midpt - k0) / (k1 - k0);
                        vlimhorz = hc[k].v;
                        }
```

```
		else if (midpt >= k2 && midpt <= k3
		 && midpt > ru_hcurv[k].s) {	/*out spiral*/
			seg.ir = 1 / ru_hcurv[k].r * (midpt - k3) / (k2 - k3);
			hc[k].seen = 'Y';
			vlimhorz = hc[k].v;
			}
		else if (midpt > k1 && midpt < k2) { /*in circle*/
			seg.ir = 1 / ru_hcurv[k].r;
			vlimhorz = hc[k].v;
			}
		else {				/*after curve k*/
			if (k < ru_nhcurv - 1) {
				++k;
				goto testwithink;
				}
			else
				seg.ir = vlimhorz = 0;
			}

		/*done with h and v curv figuring. Now set seg.vmax
		to the minimum of the two vlimits and write the record*/
		seg.vmax = (vlimvert < vlimhorz) ? vlimvert : vlimhorz;
		if (i == ru_smin)
			seg.vmax = 0;	/*must start from stopped*/
		rupwrite(&seg, rup_nrec);
		prevseg = seg;
		}

	/*write the final record (identical to second-to-last record)*/
	seg.start = seg.end;
	seg.vmax = 0;			/*must end stopped*/
	rupwrite(&seg, rup_nrec);

	close(rupfile);
	puts("Completed initial building of processed route.");

	/*report any missing curves in the output (could happen if
	the segment length is too large and it skips over short
	curves)*/
	for (i = 0; i < ru_nhcurv; ++i)
		if (hc[i].seen == 'N')
			printf("Horizontal curve at %f missed! (Set segments shorter.)\n", ru_hcurv[i].s);
	for (i = 0; i < ru_nvcurv; ++i)
		if (vc[i].seen == 'N')
			printf("Vertical curve at %f missed! (Set segments shorter.)\n", ru_hcurv[i].s);

	/*write report file*/
	if ((fo = fopen(argv[5], "wt")) == NULL) {
		puts("Can't open report file.");
		exit(3);
		}
	for (i = 0; i < ru_nhcurv; ++i) {
		fprintf(fo, "%f %f %f %f %f %f %f\n",
		ru_hcurv[i].s, ru_hcurv[i].r, ru_hcurv[i].curve,
		hc[i].Ls, hc[i].Lc, hc[i].v,
		atan(hc[i].v * hc[i].v / ru_hcurv[i].r / g));
		}
	fclose(fo);
	puts("Written report file.");
	}
```

```
#include "route.h"
#include "par.cpp"
#include "rup.cpp"

/*misc*/
int nroots;
float root[4];          /*stores intermediate roots in curve calculation*/

float min(float x, float y)
  {return (x > y) ? y : x;}

float vf(float v0, float a, float s)
  {
  float v2 = v0 * v0 + 2 * a * s;
  return v2 > 0 ? sqrt(v2) : 0;
  }

void sqrtest(float a, char *funcid, float p)
  {
  if (a >= 0)
        return;
  printf("Negative under square root found\nin calculation of %s at point %f\n",
  funcid, p);
  exit(1);
  }

void swap(float *a, float *b)
  {
  float c;

  c = *a;
  *a = *b;
  *b = c;
  }

void sillysort(float *a, float *b, float *c)
  {
  again:
  if (*a > *b)
        swap(a, b);
  if (*b > *c)
        swap(b, c);
  if (*a > *b || *b > *c)
        goto again;
  }


void findroots(float ds, float dd, float ir2, float a)
  /*finds where the vertical acceleration = f(velocity) curve
  crosses the line acceleration = a, and adds the results to
  the array root[], where nroots indicates how much is already
  in said array*/
  {
  float dd2 = dd * dd, ds2 = ds * ds;
  float p;        /*inner radical*/
  float q;

  p = g2 * dd2 / ds2
    + (a * a - g2)
    * (ir2 + dd2 / ds2);
  if (p < 0)
        return;
  p = sqrt(p);
```

```
        q = (-g * dd / ds + p)
          / (ir2 + dd2 / ds2);
        if (q >= 0)
                root[nroots++] = sqrt(q);
        q = (-g * dd / ds - p)
          / (ir2 + dd2 / ds2);
        if (q >= 0)
                root[nroots++] = sqrt(q);
        }


void limit(struct rupseg *s, float b)
  /*limits the segment s to a maximum inst velocity of b
  Only changes s->vmax1 and 2, not s->vmax*/
  {
  if (s->vmin2 == 0) {            /*there is only one velocity window*/
          s->vmax1 = s->vmax2 = min(b, s->vmax2);
          }
  else if (b < s->vmin2) {        /*b is in lower window or between wiundows*/
          s->vmax1 = s->vmax2 = min(b, s->vmax1);
          s->vmin2 = 0;           /*now there is one window 0--b*/
          }
  else if (b < s->vmax2) {        /*b is in upper window*/
          s->vmax2 = b;           /*shrink upper window*/
          }
  }

void calca2()
  /*calculates instantaneous velocity limits vmax1, vmin2, vmax2*/
  {
  long i;
  float vmost, dd, ds, ir2, v2;
  float a, c;     /*general use*/
  struct rupseg s, sprev, snext;
  float tanh = tan(hmax);
          /*tan(roll angle) is what is actually used..*/

  printf("Calculating instantaneous velocity limitations...\n");
  for (i = 1; i < rup_nrec - 1; ++i) {
          rupread(&sprev, i - 1);
          rupread(&s, i);
          rupread(&snext, i + 1);
          dd = s.d - sprev.d;             /*change in grade*/
          ds = s.end - sprev.end;         /*distance covered*/
          ir2 = s.ir * s.ir;

          /*CASE I*/
          /*if there is only a horizontal curve but no vertical*/
          if (s.ir != 0 && dd == 0) {
                  a = (amax2 * amax2 - g2) / ir2;
                  sqrtest(a, "horizontal-only acceleration limit", s.end);
                  s.vmax1 = s.vmax2 = sqrt(sqrt(a));
                  s.vmin2 = 0;
                  goto segdone;
                  }

          /*CASE II*/
          /*if there is no type of curve (h or v)*/
          if (s.ir == 0 && dd == 0) {
                  s.vmax1 = s.vmax2 = vmax;
                  s.vmin2 = 0;
                  goto segdone;
                  }
```

```
        /*CASE III*/
        /*if there is only a vertical curve but no horizontal*/
        if (s.ir == 0) {
                a = (dd > 0) ? amax2 : amin2;
                s.vmax1 = s.vmax2 = sqrt((a - g) * ds / dd);
                s.vmin2 = 0;
                goto segdone;
                }

        /*CASE IV.a */
        /*if there is a positive vertical curve (valley) and
        a horizontal curve, test for amax only*/
        if (dd > 0) {
                nroots = 0;
                findroots(ds, dd, ir2, amax2);
                if (nroots != 1) {
                        printf("!=1 roots in valley\n");
                        exit(1);
                        }
                s.vmax1 = s.vmax2 = root[0];
                s.vmin2 = 0;
                goto segdone;
                }

        /*CASE IV.b */
        /*if there is a negative vertical curve (mountain) and
        a horiontal curve, test for amin and amax.
        There can be 1, 2 or 3 roots in this case*/
        nroots = 0;
        findroots(ds, dd, ir2, amax2);
        findroots(ds, dd, ir2, amin2);
        if (nroots > 3 || nroots < 1) {
                printf(">3 or <1 root in mountain\n");
                exit(1);
                }

        /*if one root*/
        if (nroots == 1)  {
                s.vmax1 = s.vmax2 = root[0];
                s.vmin2 = 0;
                goto segdone;
                }

        /*if two roots, throw out the lesser of the two*/
        else if (nroots == 2) {
                printf("Two roots found at %ld!\n\n", s.end);
                a = (root[0] > root[1]) ? root[0] : root[1];
                s.vmax1 = s.vmax2 = a;
                s.vmin2 = 0;
                goto segdone;
                }

        /*if three roots, vmax1 is the least of these and
        set the secondary v range to the other 2*/
        else {
                printf("Three roots found at %ld!\n\n", s.end);
                sillysort(&root[0], &root[1], &root[2]);
                s.vmax1 = root[0];
                s.vmin2 = root[1];
                s.vmax2 = root[2];
                goto segdone;
                }
```

```
            segdone:

            /*now adjust velocity to the roll angle limitation*/
            if (s.ir) {
                    a =
                     tanh * g / s.ir
                     / (1 - dd * tanh / (s.ir * ds));
                    if (a > 0)
                            limit(&s, sqrt(a));
                    /*it seems to be possible to have a negative
                    a if s.ir is small and dd is positive. In this
                    case do not limit velocity*/
                    }


            /*limit vmax's to the system vmax*/
            limit(&s, vmax);

            /*if s.vmax was not already set (ie by makerup) or
            it is too high, set it to s.vmax2*/
            if (s.vmax <= 0 || s.vmax > s.vmax2)
                    s.vmax = s.vmax2;

            /*now save the results and go on to the next record*/
            rupwrite(&s, i);
            }
    puts("Done calculating.");
    }


  void main(int argc, char *argv[])
    {

    puts("Calculates allowable instantaneous velocities along maglev guideway.");

    /*evaluate parameters*/
    if (argc != 3) {
            puts("Usage: VCALC <parameter-file> <route-file>");
            exit(1);
            }

    readpar(argv[1]);
    rupopen(argv[2], 0);
    calca2();
    close(rupfile);
    }
```

```
/*traverses route in both directions and sets allowable time-
related velocities according to mech power available and
ride quality; also records power drawn and component ride
quality profile*/

#include "route.h"
#include "par.cpp"
#include "rup.cpp"

#define BANKLESS 0
  /*the number of radians less the guideway should be banked from
  the roll angle (also the allowed amt of super-roll) */
#define POWERREDUCT 0.05
  /*velocity reduction to use when power is too high*/

void calcpower(struct rupseg *s, struct rupseg *sprev)
  /*calculates w, E, and cumE in s*/
  {
  int i, wv;
  float dE, dA, d, liftdrag, weight;

  /*the power depends on which vehicles you use. If there are no
  large vehicles, use power constants for smaller ones*/
  wv = fracveh0 < 1 ? 1 : 0;

  /*vehicle weight depends on the parameter constant and vert accel*/
  /*this should rather be named "dynamic loading"*/
  weight = vehweight[wv] * s->a2 / g;

  /*find the components of drag*/
  dA = AE1[wv] * s->vmax * s->vmax;
                /*aero drag always the same*/
/*        dE = EM1[wv] * (exp(EM2[wv] * s->vmax) - exp(EM3[wv] * s->vmax));
                EM drag for mode 2 NOT USED*/
  liftdrag = LD1[wv] + LD2[wv] * s->vmax + LD3[wv] * s->vmax * s->vmax;
                /*lift to drag ratio*/

  /*depending on the speed, figure out the drag (assumes that landing
  is occuring at any time the speed goes below vmode3*/
  if (s->vmax < vmode2) {
          s->promode = 1;
          s->drag = dA + weight * skidfriction[wv];
          }
  else if (s->vmax < vmode3) {
          s->promode = 2;
          s->drag = drag2a[wv] * s->vmax + drag2c[wv];
            /*straight line fit of take-off drag*/
          }
  else {
          s->promode = 3;
          s->drag = dA + weight / liftdrag;
          }

  /*calculate the energy used in the segment: it is composed of
  power used to accelerate, overcome drag, and overcome
  gravity on grades (100% efficient for now)
          (E = (force+force+force) * distance)
  */
  s->E = (s->a1 * vehmass[wv] + s->d * g * vehmass[wv] + s->drag)
   * (s->end - sprev->end);

  /*keep track of mechanical power (energy over time)
  and cumulative energy*/
```

```
        s->w = s->E / s->time;
        s->cumE = sprev->cumE + s->E;
        }

    float vf(float v0, float a, float s)
        {
        float v2 = v0 * v0 + 2 * a * s;
        return v2 > 0 ? sqrt(v2) : 0;
        }

    float absF(float a)
        {
        return a > 0 ? a : -a;
        }

    void limit(struct rupseg *s, float b)
        /*limits the segment s to a maximum velocity of b*/
        {
        if (s->vmax <= b)               /*if already less than b, do nothing*/
                return;
        if (b <= 0) {
                printf("Tried to set a nonpositive velocity at %f.\n", s->start);
                exit(1);
                }
        s->vmax = b;
        if (b < s->vmin2)               /*if between velocity windows*/
                s->vmax = s->vmax1;     /*reduce to lower window*/
        }

    void calcVLR(struct rupseg *s, struct rupseg *sadj, int bwflag)
        /*calculates these fields in *s :
           roll, bank, time, cumtime, a1, a2, and a3.
        Bidirectional:
        * If calling in a forward scan, sadj is the record before,
                bwflag=0
        * If calling in a backward scan, sadj is the next record,
                bwflag=1, times and a1 are not calculated
        */
        {
        float a, c, v2;
        float dd = bwflag ? sadj->d - s->d : s->d - sadj->d;
        float ds = s->end - s->start;

        /*ASSUME COORDINATED TURNS:
        calculate vertical acceleration and
        roll angle and lateral acceleration at s.vmax*/
        v2 = s->vmax * s->vmax;
        a = v2 * s->ir;
        c = dd * v2 / ds + g;
        s->a2 = sqrt(a * a + c * c);
        s->a3 = 0;
        s->roll = atan(a / c);

        /*guideway bank is some # less than roll angle*/
        s->bank = s->roll - BANKLESS;
        if (s->bank < 0)
                s->bank = 0;

        /*calculate the time and fwd acceleration in the seg
        calculated from midpoints*/
        if (!s->vmax) {
                printf("Zero vmax found at %ld\n", s->end);
                exit(1);
```

```
        }
    if (bwflag == 0) {
            s->time = (s->end - s->start) / s->vmax;
            s->cumtime = sadj->cumtime + s->time;
            s->a1 = (s->vmax - sadj->vmax) / ((s->time + sadj->time) * 0.5);
            }

    return;
    }

/*functions calca1b and calca1*/
  /*calculate velocity limit vmax using time-related variables:
  - can't go faster than what you need to slow down for future
    slowdowns
  - can't go faster than what you could accelerate from the previous
    speed in the given time

  calculate acclerations, roll, bank, time, and power

  - can't go faster than what would violate other acceleration
    or power limits.
  */


void calca1b()
  /*backwards calculation*/
  {
  long i;
  float vmost, a;
  struct rupseg s, sprev, snext, sprevprev;

  puts("Calculating deceleration requirements.");
  for (i = rup_nrec - 2; i > 0; --i) {
          rupread(&sprev, i - 1);
          rupread(&s, i);
          rupread(&snext, i + 1);

          calcVLR(&s, &snext, 1);
          if (s.vmax <= snext.vmax)
                  continue;

          /*limit the velocity to what could be achieved
          at abs(max deceleration) starting at midpt of the
          next segment and going backwards to the midpt of this
          segment.  Deceleration is limited as below for the calc
          of max accleration*/
          a = s.a2 - g;
          if (a > amax123)
                  a = 0;
          else
                  a = sqrt(amax123 * amax123 - a * a);
          if (a > -amin1)
                  a = -amin1;
          vmost = vf(snext.vmax, a, (snext.end - sprev.end) * 0.5);
          limit(&s, vmost);

          rupwrite(&s, i);
          }
  }

void calca1()
  /*forwards calculation*/
  {
```

```
  long i;
  float vmost, dd, ds, v2, a, c;
  struct rupseg s, sprev, snext;

  puts("Calculating acceleration/power requirements.");
  for (i = 1; i < rup_nrec - 1; ++i) {
          rupread(&sprev, i - 1);
          rupread(&s, i);
          rupread(&snext, i + 1);

          calcVLR(&s, &sprev, 0);

          /*limit the velocity to what could be
          achieved at maximum acceleration starting at the midpt of
          the previous segment going to the midpt of this segment.
          Maximum acceleration is amax1 possibly reduced by the
          existence of a vertical acceleration, which would
          together exceed amax123*/
          a = s.a2 - g;
          if (a > amax123)
                  a = 0;
          else
                  a = sqrt(amax123 * amax123 - a * a);
          if (a > amax1)
                  a = amax1;
          vmost = vf(sprev.vmax, a, (s.end - sprev.start) * 0.5);
          limit(&s, vmost);
          calcVLR(&s, &sprev, 0);

          /*calculate power and energy used*/
          calcpower(&s, &sprev);
          if (s.w > maxpower) {
                  printf("overpower at %ld, speed %f --> ", s.start, s.vmax);
                  do { /*go down bit by bit until ok*/
                          limit(&s, s.vmax - POWERREDUCT);
                          calcVLR(&s, &sprev, 0);
                          calcpower(&s, &sprev);
                          } while (s.w > maxpower);
                  printf("%f\n", s.vmax);
                  }

          rupwrite(&s, i);
          }
  }

void main(int argc, char *argv[])
  {

  puts("Calculates real allowable velocities along maglev guideway.");

  /*evaluate parameters*/
  if (argc != 3) {
          puts("Usage: VCALCT <parameter-file> <route-file>");
          exit(1);
          }

  readpar(argv[1]);
  rupopen(argv[2], 0);

  calca1b();
  calca1();

  close(rupfile);
```

```
    puts("Done.");
    }
```

```
#include "route.h"
#include "ru.cpp"
#include "rup.cpp"

#define TODEG (180/3.141592654)

void stats()
  {
  printf("Total distance of route:    %f\n", smax);
  printf("Total transit time:         %f\n", tmax);
  if (tmax)
  printf("Average velocity:           %f\n", smax / tmax);
  printf("Total energy for 1 trip:    %f\n", Emax);
  }

void main(int argc, char *argv[])
  {
  int i;
  struct rupseg s;
  FILE *fp;
  float rolldeg, bankdeg;

  /*evaluate parameters*/
  if (argc != 4) {
        puts("Usage: RUPREP <input-file> <output-file> <report-type>");
        puts("       report-type: 1=ride quality  2=guideway  3=copmlete");
        exit(1);
        }

  if ((fp = fopen(argv[2], "wt")) == NULL) {
        puts("Can't open output file.");
        exit(3);
        }

  rupopen(argv[1], 0);
  rupset();
  stats();
  puts("Writing output file...");
  for (i = 0; i < rup_nrec; ++i) {
        rupread(&s, i);

        /*convert radians to degrees*/
        rolldeg = s.roll * TODEG;
        bankdeg = s.bank * TODEG;

        switch(argv[3][0]) {
              case '1': /*ride quality*/
              fprintf(fp,
              "%ld %ld %d %e %e %e %e %e %e %e %e %e\n",
              s.start, s.end, s.promode, s.ir, s.d, s.vmax,
              s.time, s.cumtime, s.a1, s.a2, s.a3, rolldeg);

              break;

              case '2': /*guideway*/
              fprintf(fp,
              "%ld %ld %f %d %e %e %e %e %e %e %e\n",
              s.start, s.end, bankdeg, s.promode, s.ir, s.d, s.vmax,
              s.a1, s.a2, s.a3, rolldeg);
              break;

              case '3': /*complete*/
              fprintf(fp,
```

```
                    "%ld %ld %f %d %e %e %e %e %e %e %e %e %e %e %e %e %e\n",
                    s.start, s.end, bankdeg, s.promode, s.ir, s.d, s.vmax,
                    s.time, s.cumtime, s.w, s.E,
                    s.cumE, s.drag, s.a1, s.a2, s.a3, rolldeg);
                    break;
                    }
            }
    close(rupfile);
    fclose(fp);
    puts("Written.");
    }
```

```
#include "route.h"
#include "mem.h"
#include "rup.cpp"
#include "par.cpp"

#define MAXBLOCKS 1000

int nveh0, nveh1; /*number of vehicles*/
int nblocks;            /*number of blocks*/
float vehperiod;  /*time between vehicles*/

typedef struct blk_s {
  long rec1, rec2; /*start and end rup records*/
  float start, end;/*endpoint*/
  float wmin, wmax;/*utility power required*/
  float E;         /*utility energy drawn for one veh pass*/
  float cost[2];   /*all costs that vary with block length*/
  float tcost;     /*life cycle cost*/
  int type;        /*0=normal, 1=half R LSM*/
  } block;

block b[MAXBLOCKS];


float timetos(float t)
  /*returns the distance s which is reached at time t.
  Finds this by searching route*/
  {
  float frac;
  struct rupseg s, sprev;
  static long rec = 1;

  while (1) {
          rupread(&sprev, rec - 1);
          rupread(&s, rec);
          if (s.cumtime >= t && sprev.cumtime < t) {
                  frac = (t - sprev.cumtime) / (s.time);
                  return (s.end - sprev.end) * frac + sprev.end;
                  }
          else if (s.cumtime < t && rec < rup_nrec)
                  ++rec;
          else if (sprev.cumtime > t && rec > 1)
                  --rec;
          else
                  break;
          }

  printf("Invalid value for time found.\n");
  return 0;
  }

int makeblock(block *b, long rec1, long rec2)
  /*returns 1 on success else 0. Can fail if block is longer than
  vehperiod or 2 km*/
  {
  long rec;
  struct rupseg s, slast;
  float I, P, R, T, Eutil, Putil, Ploss, wabsmax, Tmax;

  /*initialize block*/
  rupread(&slast, rec2);
  rupread(&s, rec1);
  b->rec1 = rec1;
```

```
        b->rec2 = rec2;
        b->end = slast.end;
        b->start = s.start;
        b->E = 0;
        b->wmin = 1e9;
        b->wmax = -1e9;
        b->cost[0] = b->cost[1] = b->tcost = 0;
        b->type = 0;

        /*if too long bailout*/
        if (b->end - b->start > 2000
         || slast.cumtime - s.cumtime > vehperiod)
                return 0;

        /*step through segments and find maximum thrust in block*/
        Tmax = 0;
        for (rec = rec1; rec <= rec2; ++rec) {
                rupread(&s, rec);
                T = s.w / s.vmax;
                if (T > Tmax)
                        Tmax = T;
                }

        /*If thrust is high anywhere in this block, use low-R LSM
        in the whole block. Calculate R (resistance/block/phase)*/
        if (T < thrustthresh)
                R = Rperm * (b->end - b->start);
        else {
                R = Rperm / 2 * (b->end - b->start);
                b->type = 1;
                }

        /*step through segments*/
        for (rec = rec1; rec <= rec2; ++rec) {
                rupread(&s, rec);

                /*calc thrust, current, and LSM power loss*/
                T = s.w / s.vmax;
                I = T * ItoF;
                Ploss = 3 * I * I * R;
                Ploss += Photel[1];     /*cast as LSM loss although it isn't really
                                        but works out just as well*/

                /*calc utility power gain or loss and utility energy*/
                if (s.w > 0) {
                        Putil = peff * (s.w + Ploss);
                        }
                else {
                        Putil = peffback * (s.w - Ploss);
                        }
                Eutil = Putil * s.time;
                b->E += Eutil;

                /*store the min/max utility power for the block*/
                if (b->wmax < Putil)
                        b->wmax = Putil;
                if (b->wmin > Putil)
                        b->wmin = Putil;

                /*add the cost of electricity used in this segment
                (per average second). This is the cost of electricity
                used (Eutil) times the duty cycle*/
                b->cost[1] += costW[1] * Eutil * s.time / vehperiod;
```

```
		}

	/*find the other costs, ignoring those that do not affect
	block optimization*/

	/*cost for Low-R LSM*/
	if (b->type)
		b->cost[0] += costlowR[0] * (b->end - b->start);

	/*cost for the block in general and for the power rating*/
	wabsmax = (abs(b->wmax) > abs(b->wmin)) ? b->wmax : b->wmin;
	b->cost[0] += costb[0];
	b->cost[1] += costb[1];
	b->cost[0] += costbW[0] * wabsmax;
	b->cost[1] += costbW[1] * wabsmax;

	/*total costs*/
	b->tcost = b->cost[0] + b->cost[1] * opttime;
	return 1;
	}

void makeblocks()
	{
	long rec1, rec2, bestendrec;
	float bestcost, thiscost;	/*costs per meter to find optimal block*/

	puts("Optimizing block lengths...");

	/*initial conditions*/
	rec1 = 1;
	rec2 = 1;
	nblocks = 0;

	makenextblock:

	printf("- %d ", nblocks + 1);
	bestendrec = -1;
	bestcost = 1e99;

	/*try out every block length, storing them one past the end
	of the b array*/
	while (1) {
		if (!makeblock(&b[nblocks], rec1, rec2))
			goto stopcomparing;
		thiscost = b[nblocks].tcost / (b[nblocks].end - b[nblocks].start);
		if (thiscost < bestcost) {
			bestendrec = rec2;
			bestcost = thiscost;
			}
		if (++rec2 >= rup_nrec - 1)
			goto stopcomparing;
		}

	stopcomparing:

	if (bestendrec < 0) {
		puts("\nMin cost never found.");
		exit(1);
		}
	makeblock(&b[nblocks], rec1, bestendrec);  /*do the best one again*/
	if (++nblocks >= MAXBLOCKS) {		/*include in array*/
		puts("No more room to store blocks");
		exit(1);
```

```
                }

        /*do the next block if there are any more*/
        if (rec2 < rup_nrec - 1) {
                rec1 = rec2 = bestendrec + 1;
                goto makenextblock;
                }

        puts("blocks.");
        }


void main(int argc, char *argv[])
        {
        long i, c, pcap;
        float Eutiltot, Wutilmin, Wutilmax;
        FILE *fo;

        puts("Maglev guideway cost optimizer.");

        /*evaluate parameters*/
        if (argc != 4) {
                puts("Usage: COSTOPT <parameter-file> <route-file> <output-file>");
                exit(1);
                }

        readpar(argv[1]);
        rupopen(argv[2], 0);
        rupset();

        if ((fo = fopen(argv[3], "wt")) == NULL) {
                puts("Can't open output file.");
                exit(3);
                }

        if (tmax <= 0) {
                puts("Total time is zero: cannot optimize.");
                exit(1);
                }
        pcap = pps * tmax + 1;                  /*required route capacity*/

        /*calculate number of vehicles - the .99 insures an extra
        vehicle for people left over by the division*/
        nveh0 = fracveh0 * pcap / vehcapacity[0] + .99;
        nveh1 = (1 - fracveh0) * pcap / vehcapacity[1] + .99;

        /*calc the time between vehicles (period)*/
        vehperiod = tmax / (nveh0 + nveh1);

        /*make blocks*/
        makeblocks();

        /*calc total energy drawn from utility & stuff*/
        Eutiltot = Wutilmax = Wutilmin = 0;
        for (i = 0; i < nblocks; ++i) {
                Eutiltot += b[i].E;
                if (Wutilmax < b[i].wmax) Wutilmax = b[i].wmax;
                if (Wutilmin > b[i].wmin) Wutilmin = b[i].wmin;
                }

        /*tell the user stuff*/
        printf("Mech energy = %e\n", Emax);
        printf("Util energy = %e\n", Eutiltot);
```

```
printf("Vehicles: %d small and %d large\n", nveh0, nveh1);
printf("          %f s between\n", vehperiod);
printf("Blocks: %d\n", nblocks);
printf("Util power range = %e - %e\n", Wutilmin, Wutilmax);

/*outputs*/
puts("Writing output file...");
fprintf(fo, "%e\n%e\n%e\n", smax, tmax, Emax);
fprintf(fo, "%d\n%d\n%d\n", nveh0, nveh1, nblocks);
fprintf(fo, "%e\n%e %e\n%e\n", Eutiltot, Wutilmin, Wutilmax, vehperiod);
for (i = 0; i < nblocks; ++i) {
        fprintf(fo, "%e %e %e %e %e %e %e %d\n",
        (float)i, b[i].end, b[i].wmin, b[i].wmax, b[i].E, b[i].cost[0],
         b[i].cost[1], b[i].type);
        }
fclose(fo);
puts("Done.");
}
```

```
/*
reverses a ru file
parameters are input-file output-file
*/

#include "route.h"

#define RU_MAXCURV 200

float ru_smin, ru_smax;                 /*endpoints of route*/
int ru_nhcurv, ru_nvcurv;              /*number of curves*/

struct {
  float s, r;      /*point, radius*/
  float curve;     /*radians*/
  } ru_hcurv[RU_MAXCURV];         /*list of horiz curves*/

struct {
  float s, L;      /*point, curve length*/
  float d1, d2;    /*grade before and after*/
  } ru_vcurv[RU_MAXCURV];         /*;ost of vert curves*/

void readru(char *name)
  /*reads in given file into above variables. DIFFERENT FROM
  THE VERSION OF THIS IN RU.CPP: has fewer checks and does not
  cvt to radians*/
  {
  FILE *fi;
  char buf[80], *buf2;
  int i;

  if ((fi = fopen(name, "rt")) == NULL) {
          puts("Can't read input file.");
          exit(3);
          }

  ru_nhcurv = ru_nvcurv = 0;
  while (fgets(buf, 80, fi) != NULL) {
          buf2 = buf + 1;
          switch (toupper(buf[0])) {
                  case 'S':
                  if (ru_smax) {
                          puts("Total distance reported twice.");
                          exit(6);
                          }
                  sscanf(buf + 1, "%f%f", &ru_smin, &ru_smax);
                  if (ru_smin < 0 || ru_smax <= ru_smin) {
                          puts("Total distance is negative or zero.");
                          exit(5);
                          }
                  break;

                  case 'H':
                  if (ru_nhcurv >= RU_MAXCURV) {
                          puts("Too many horizontal curves.");
                          exit(10);
                          }
                  sscanf(buf2, "%f%f%f", &ru_hcurv[ru_nhcurv].s,
                   &ru_hcurv[ru_nhcurv].r, &ru_hcurv[ru_nhcurv].curve);
                  if (ru_hcurv[ru_nhcurv].s < 0
                   || ru_hcurv[ru_nhcurv].r <= 0
                   || ru_hcurv[ru_nhcurv].curve <= 0
                   || ru_hcurv[ru_nhcurv].curve >= 360) {
```

```
                                printf("Invalid horizontal curve (s = %f).\n",
                                 ru_hcurv[ru_nhcurv].s);
                                exit(7);
                                }
                        ++ru_nhcurv;
                        break;

                        case 'V':
                        if (ru_nvcurv >= RU_MAXCURV) {
                                puts("Too many vertical curves.");
                                exit(10);
                                }
                        sscanf(buf + 1, " %f %f %f %f", &ru_vcurv[ru_nvcurv].s,
                         &ru_vcurv[ru_nvcurv].L, &ru_vcurv[ru_nvcurv].d1,
                         &ru_vcurv[ru_nvcurv].d2);
                        if (ru_vcurv[ru_nvcurv].s < 0
                         || ru_vcurv[ru_nvcurv].L <= 0) {
                                printf("Invalid vertical curve (s = %f).\n",
                                 ru_vcurv[ru_nvcurv].s);
                                exit(8);
                                }
                        ++ru_nvcurv;
                        break;

                        otherwise:
                        puts("Unrecognized letter in column 0.");
                        exit(9);
                        }
                }
        puts("Read route file (.ru)");
        }

void main(int argc, char *argv[])
        {
        FILE *fo;
        int i;

        /*evaluate parameters*/
        if (argc != 3) {
                puts("Usage: REVERSE <input-file> <output-file>");
                exit(1);
                }

        /*get input file*/
        readru(argv[1]);

        puts("Writing reversed route...");

        /*open output file*/
        if ((fo = fopen(argv[2], "wt")) == NULL) {
                puts("Can't open output file.");
                exit(3);
                }

        /*write output file: the points are reversed, and the grades
        are inverted (incoming <-> outgoing)*/
        fprintf(fo, "s %f %f\n", ru_smin, ru_smax);
        for (i = ru_nhcurv - 1; i >= 0; --i)
                fprintf(fo, "h %f %f %f\n", ru_smax - ru_hcurv[i].s,
                ru_hcurv[i].r, ru_hcurv[i].curve);
        for (i = ru_nvcurv - 1; i >= 0; --i)
                fprintf(fo, "v %f %f %f %f\n", ru_smax - ru_vcurv[i].s,
                ru_vcurv[i].L, ru_vcurv[i].d2, ru_vcurv[i].d1);
```

```
    puts("Done.");
}
```

SUPPLEMENT H: HEAVE DAMPING CAPABILITY ANALYSIS

CONVERTER HEAVE DAMPING REQUIREMENTS

In addition to thrust, the LSM develops vertical forces which are used for damping the natural heave motion of the Magneplane vehicle. The force components depend on the thrust angle α as shown in Figure 1.

Heave damping requires that the wayside power converter change α in response to commands from wayside control equipment. Accelerometers and other sensors on the vehicle are used with the converter in a closed loop control system to damp heave motion. The overall response needs to support a closed loop bandwidth of 2-5 Hz to coordinate properly with the natural frequencies of the vehicle body.

Preliminary simulation studies were conducted to address two questions relative to heave damping requirements: (1) can the converter phase LSM make dynamic changes in α consistent with the bandwidth target and (2) does the converter rating need to be increased to support the control loop requirements.

Simulation studies were conducted using the per-phase equivalent circuit model of the LSM shown in Figure 2. E_1 is the converter voltage, E_2 is the voltage induced in the LSM winding by the vehicle propulsion magnets (the back EMF) and I is the LSM current.

Steady state operation of the compensated LSM is shown in the phasor diagram Figure 3. Notice that the angle of the small phasor $E_1 - E_2$ is α, the angle to be controlled for heave damping. But α can only be controlled by δ, the phase angle of the converter voltage E_1. The relative lengths of E_1 and E_2 are such that small changes in δ to cause larger changes in α. This apparent "gain" increases with speed and is tabulated in Table 1.

Another observation from the phasor diagram is that current increases with α due to the increase in the magnitude of $E_1 - E_2$. This is helpful since the current needs to be increased whenever the magnitude of α is increased in order to create a levitation force while maintaining constant thrust. Figure 4 shows the current and α as a function of δ at 150 m/s. The natural current increase due to circuit operation nearly matches that needed for constant thrust.

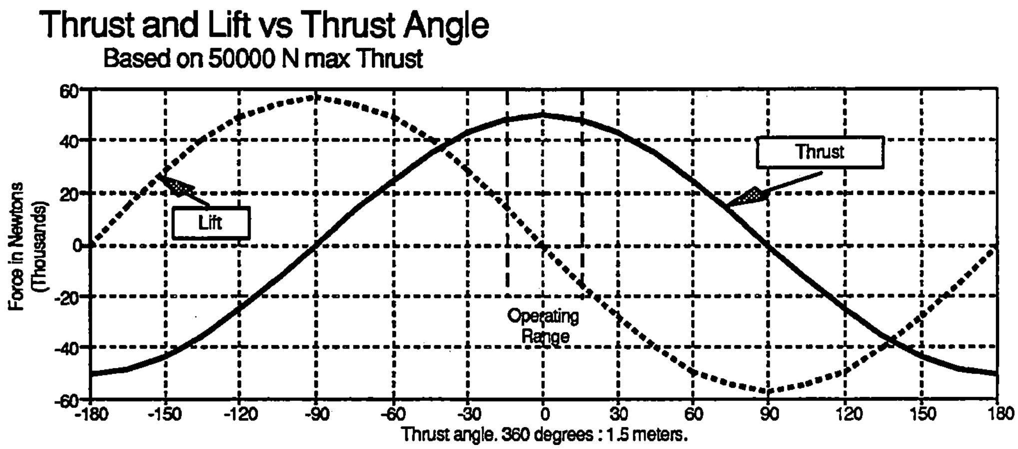
The block diagram of the simulation model is shown in Figures 5 through 8. The model is based on the per-phase equivalent circuit and assumes compensated LSM operation. Converter dynamics were not included as these should be well above the frequency range of interest. The vehicle speed was assumed to be constant at 150 m/s while phase angle changes in current (measured in terms of α) were made.

The open loop phase angle response of the LSM to a one degree change in δ is shown in Figure 9. The ultimate change in α is about 11.7 degrees as predicted from the steady state analysis and the time constant is about 0.15 seconds, corresponding to a frequency response of about 1 Hz.

A simple phase angle control loop was constructed around the LSM model to investigate dynamic response more closely. The control loop alters the phase of the converter voltage in response to phase errors while keeping the magnitude of the voltage constant. Simulation results are shown in Figure 10. In this test a 10 degree phase change was commanded and the phase response and converter current were observed. The results show an effective bandwidth between 2 and 3 Hz can be easily achieved. The maximum converter current during this response is within 5% of the steady state value.

Simulation results presented here show that the converter and series compensated LSM can support the heave damping requirements presented in this report. In addition, wayside power converter ratings should not need to be significantly increased to support operation in a closed loop heave damping system.

Figure 1. Propulsion System Forces - 140 Passenger vehicle



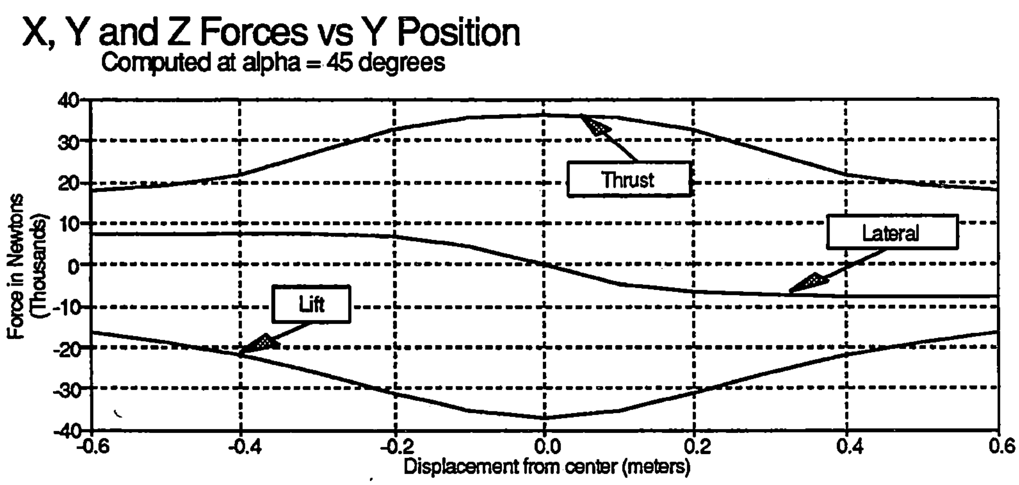


Figure 2. Per-phase Equivalent Circuit of the Compensated LSM

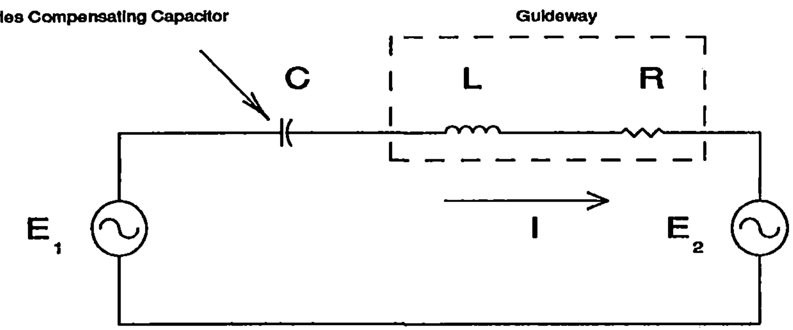


Figure 3. Phasor Diagram of the Compensated LSM

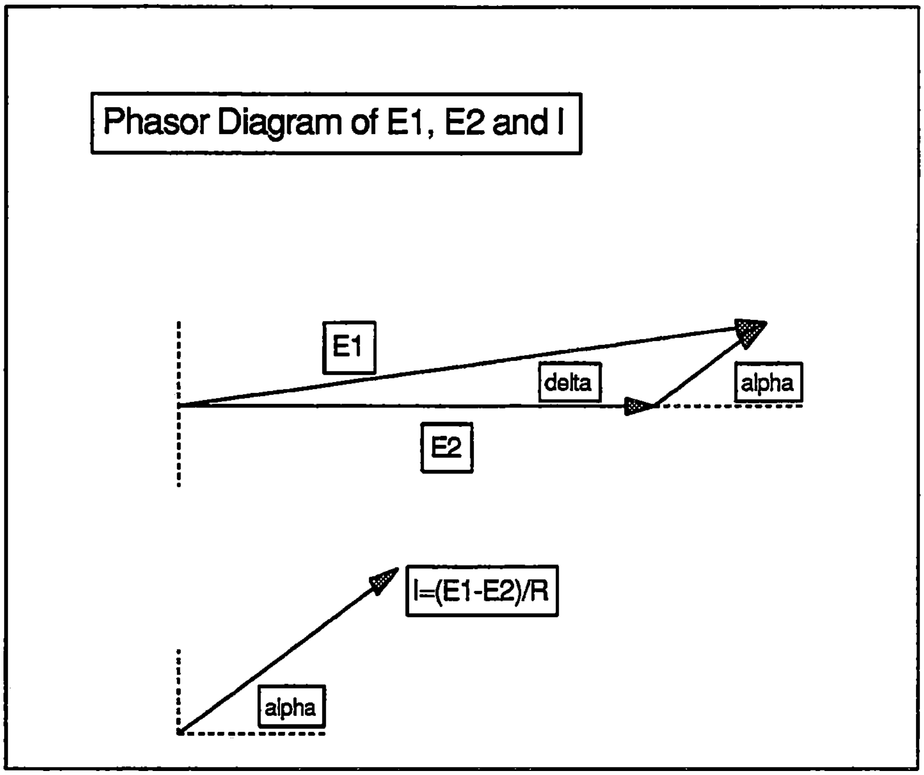


Table 1. Alpha / Delta vs Speed in Steady State

Speed	E2	E1	Alpha/Delta
0	0	215	1.0
50	775	990	4.6
100	1551	1766	8.2
150	2326	2541	11.7

Figure 4. Effect of Varying Converter Operating Angle

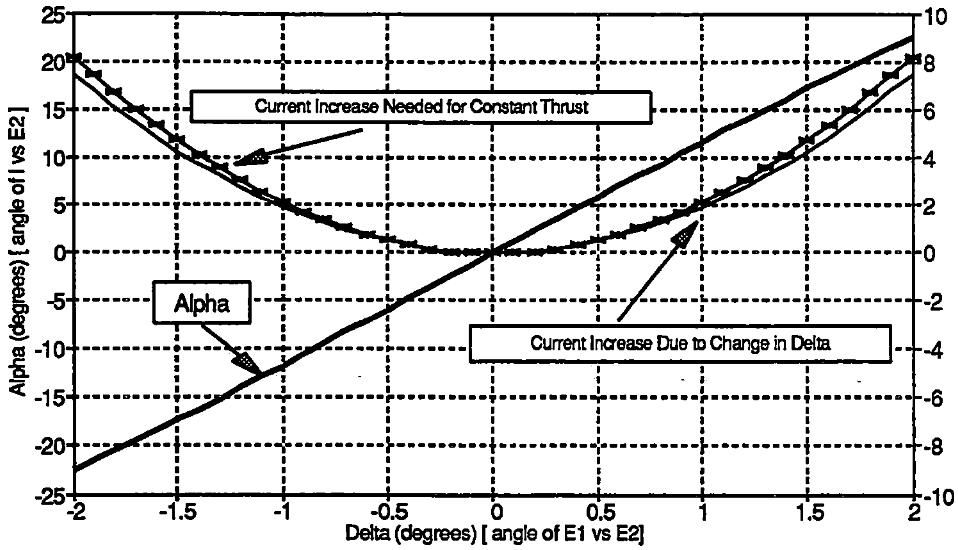


Figure 5. LSM simulation block diagram - functional diagram

(configured for open-loop response test)

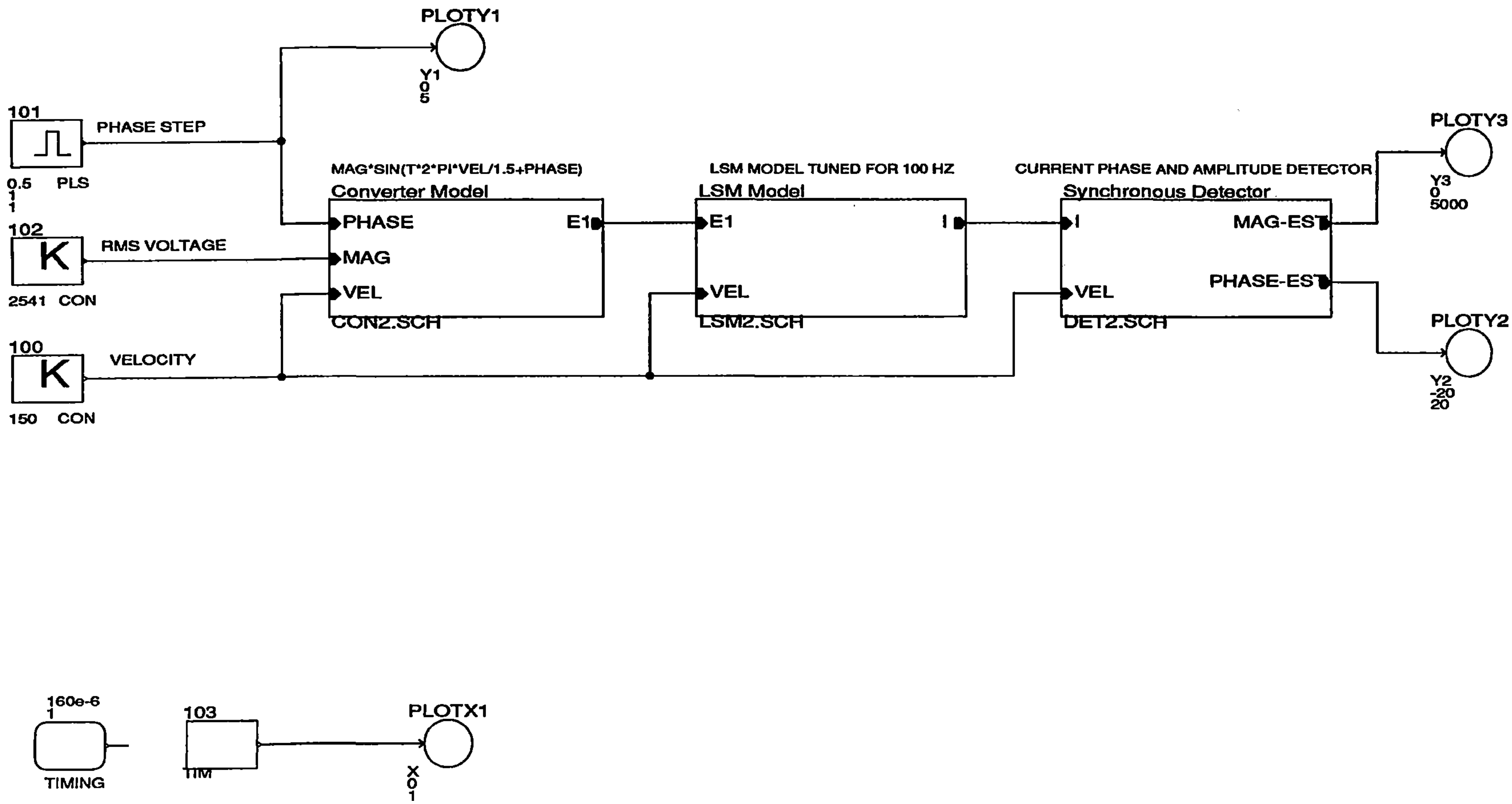


Figure 6. LSM simulation block diagram - Converter Voltage Model

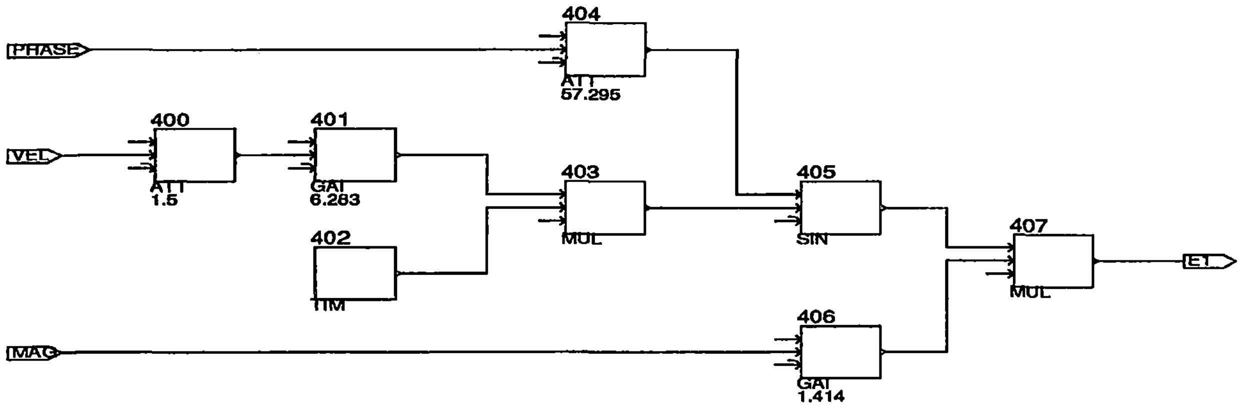


Figure 7. LSM simulation block diagram - Compensated LSM model

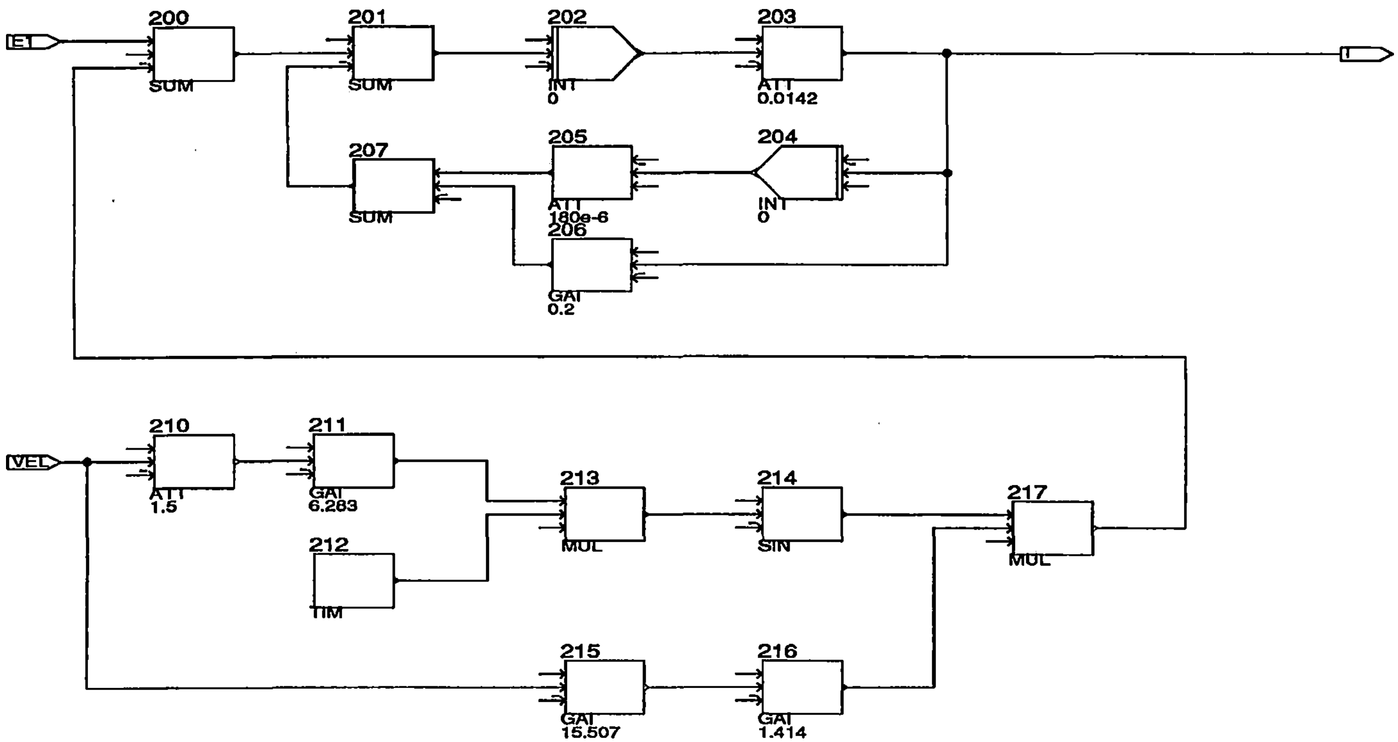


Figure 8. LSM simulation block diagram - phase /amplitude detector

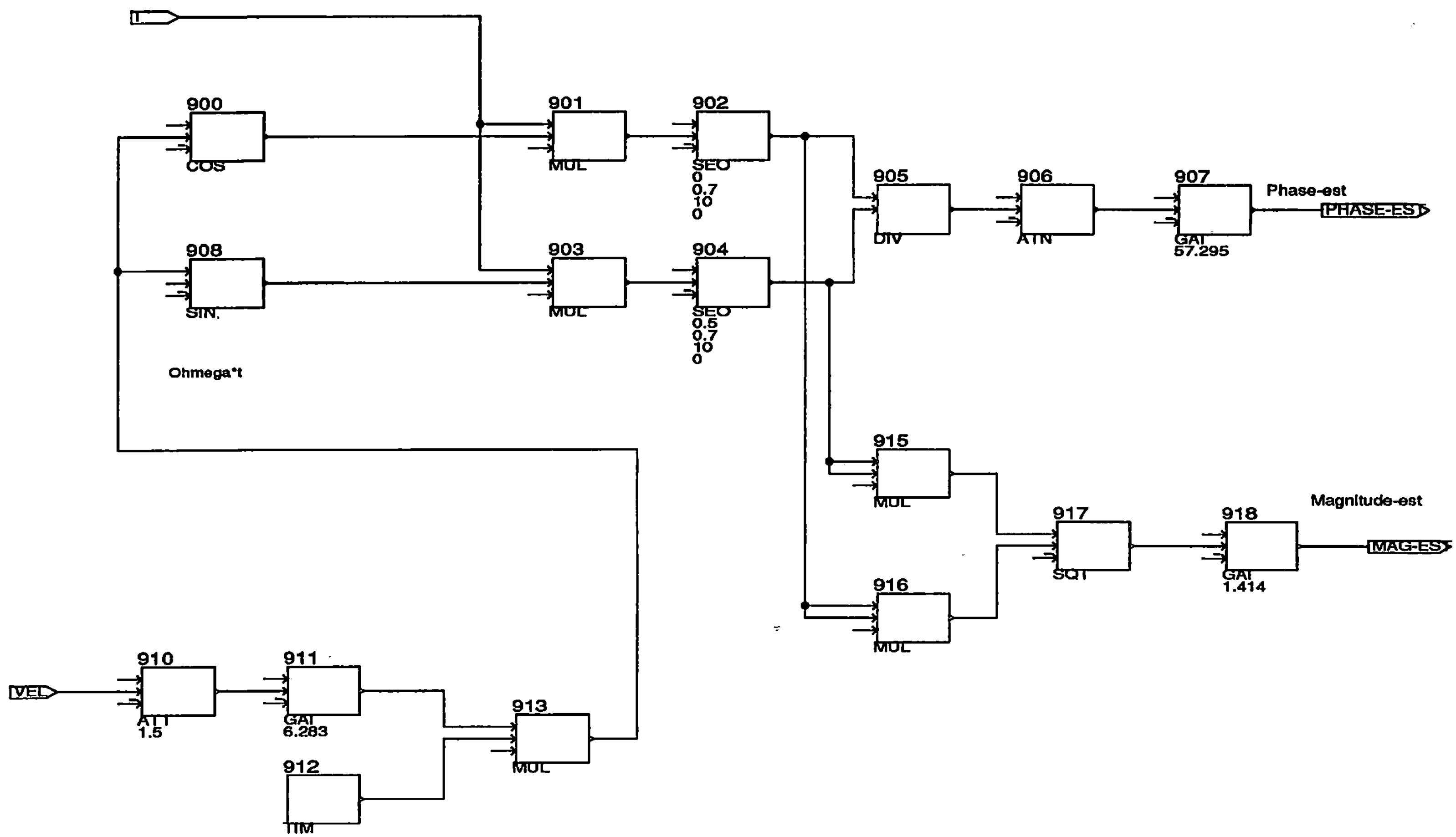


Figure 9. LSM Open Loop Phase Response

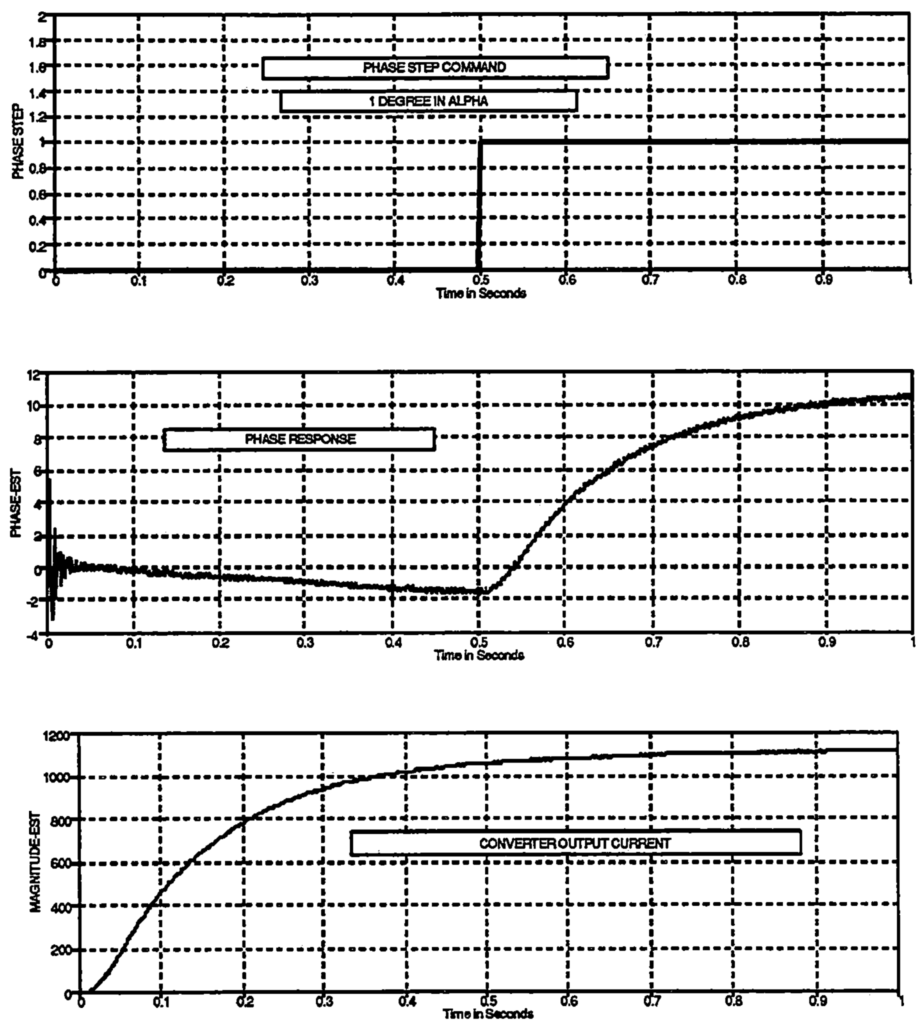
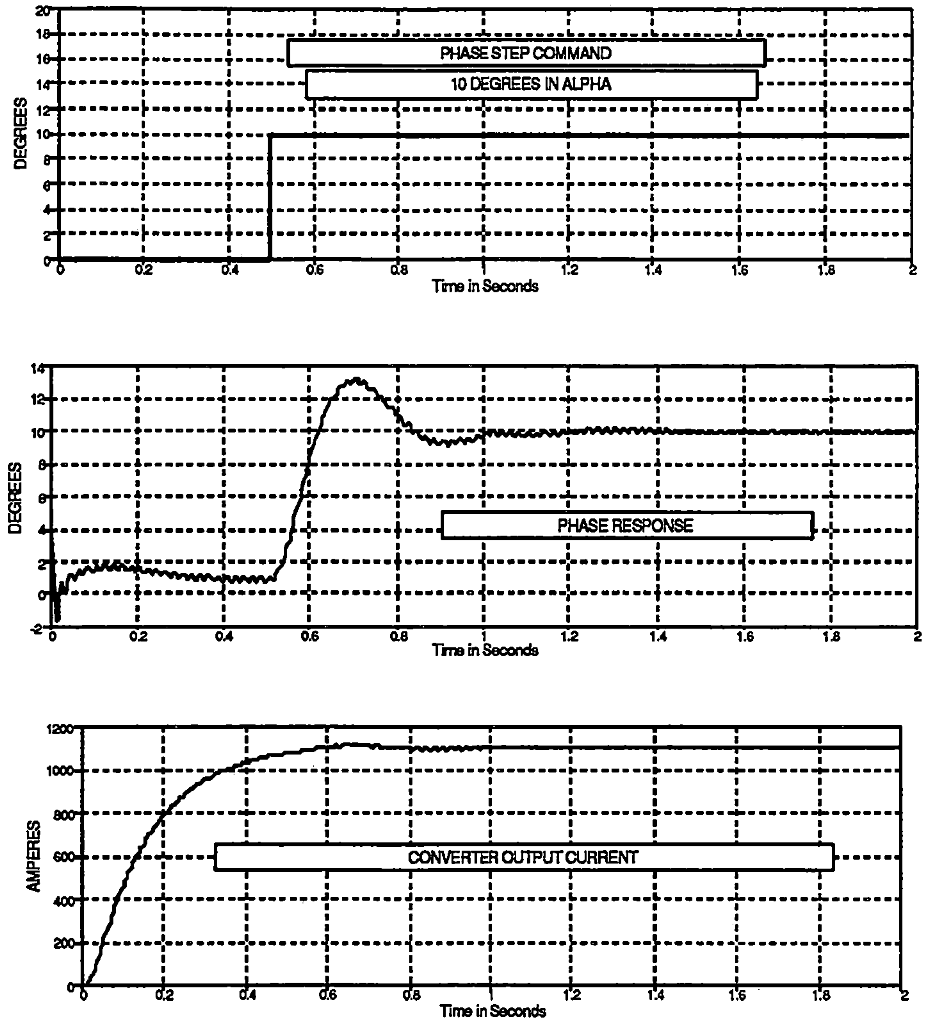


Figure 10. LSM Closed Loop Phase Response



SUPPLEMENT I: BACKUP MATERIAL FOR CONTROL AND COMMUNICATION

CONTENTS

SECTION 1. POSITION/VELOCITY MEASUREMENTS

ref. Para. 3.2.1.K.6 Position/Velocity Measurements.

(a) Position Marker (Guideway) Spacing Analysis

The positioning or separation distance (spacing) for the guideway markers is based upon complying to the following design criteria;

- Ride quality requires a typical 0.1G per axis
- Max. vehicle velocity = 150 meter/sec.
- Vehicle response characteristic ≈ 3 Hz.

Using the expression;

EQ. 1

$$\left.\frac{dv}{dt}\right|_{max} = \frac{v_{max}}{2^N \, T\,convert} = LSB$$

where

$$\left.\frac{dv}{dt}\right|_{max} = velocity\ change\ per\ time = acceleration$$

$$V_{max} = \text{max.}\ velocity\ range$$

$$N = number\ of\ samples$$

$$T\,convert = time\ required\ to\ system\ to\ respond\ to\ a\ sample\ (LSB)$$

Then converting 0.1G to meters/sec^2 = 0.1 (9.81) = 0.981 m/s^2

Using the Nyguist Theory of two (2) samples required per measurements;

Then the sample size (LSB) = 0.981/2 ≈ 0.49 m/s^2

Calculating the vehicle response time using

$$t_{response} = 0.35/BW_{-3\ db} \qquad \text{EQ. 2}$$

where vehicle BW = 2Hz

then $t_{response} = 0.35/2 = 0.18$ sec.

Substituting in EQ. 1 and solving for "N".

$$0.49 = 150/2^N\,(0.18)$$

$$2^N = 2551$$

$$N = 10.77 \approx 11$$

and $2^{11} = 2048$

Velocity (LSB) to be detected = 150 m/sec/2048 = 0.073 m/s

The marker spacing/location = $S + \Delta S$

where ΔS = total spacing error = survey error + thermal error + time base error + processing (detect/computer) EQ. 3

and survey error = 1 inch in 100 ft = 0.0008

thermal error = 2 in/70ft = 0.0024

Computation error using centroid detection and processing of the received 23.3 usec code burst based on the following signal characteristics;

```
 Code Burst #1       Code Burst #2            Code Burst #N

 _||||||____________||||||__ _ _ _ _ _ _ _ _ _   _||||||___

--->|23.2|<------46.4---->|
     us          us
```

Then processing detection error $\approx$ ½ (23.2μsec) $\approx$ 11.6μsec

ΔS for 150m/s = 150 X 11.6 X 10^{-6} = 0.002 m

$$Thermal\ error = \frac{\Delta S}{Guideway\ Section/°C} = \frac{2in}{70\ ft/60°} = 0.002$$

Spacing error due to time base accuracy (Δt) of 100 ppm is;

vehicle velocity = 2 X pole pitch X frequency

where pole pitch = 0.75 meter.

then $\Delta v = 2(0.75)\ \Delta f = 1.5\ (ppm) = 150\ ppm$

Substituting into EQ. 3 yields;

$$\Delta S = (0.0008 + 0.0024 + 0.002 + 0.00015)\ S \approx 0.005\ S\ meters$$

$$Then\ \Delta V_{LSB} = \frac{\Delta S}{T + \Delta t} = \frac{0.005\ S}{1 + 0.0001} \qquad Equasion 4$$

with T = 1 sec.

Solving for S when $\Delta V = LSB = 0.073$ m/s

$$0.073 = \frac{0.005\ S}{1 + 0.0001}$$

S = 14.6 meters

Another system criteria being fault tolerance for safety concerns advocates the smallest vehicle (22 meter length) sense at least two (2) successive markers for redundancy;

$$Therefore:\ S\ min = \frac{min.\ vehicle\ length}{2} = \frac{22}{2} = 11\ meters$$

Therefore; a minimum guideway position marker spacing of 11 meters is selected, which also satisfies the minimum system detection/control criteria.

(b) Guideway Position Marker RF Link Budget Analysis:

The transmission signal format selected consists of a code bracket pair encompassing fourteen pulse code positions. Other code characteristics are as follows;

Pulse Spacing = 1.45μsec
Pulse rise time (t_r) = 0.1μsec.
Total pulses = 16
Code combinations = 2^{14} = 16,384 plus brackets
Transmission PRF = 14.37 KHZ

Bandwidth (BW) required for t_r = 0.1μsec is;

$t_{response} = 0.35/BW_{-3\ db}$

$0.1 \times 10^{-6} = 0.35/BW$ and BW = 3.5 MHZ.

Next, the RF signal level margin for the overall communication link can be calculated using the following criteria;

XTMR ANT. GAIN	= 0db
RCVR ANT. GAIN	= 0db
RCVR Cable Loss	= 2.0 db
XTMR Cable Loss	= 2.0 db

Free Space loss (L) = 32.44 + 20 log D + 20 log F EQ. 5

when D is the distance in kilometers (Km)

F is the frequency in MHZ.

Substituting D = 0.003Km and F = 2.0GHZ in EQ. 5

$L = 32.44 + 20 \log (0.003) + 20 \log (2 \times 10^3)$

$= 98.46$ db.

The link RF budget can then be summarized as follows;

Link RF Budget (L-Band)

Transmit Pwr = 1 watt	=	30.0 dbm	
Transmit Ant. Gain & Cable Loss	=	2.0 db	
Free Space Loss	=	-98.46 db	
Actual RF Receiver Signal level	-70.46 dbm		
Rcvr. Antenna Gain			- 2.0 db

and Line Loss			
Rcvr. Sensitivity		-174.0 dbm	
BW = 3.4 MHZ		65.5	
		-108.5 dbm	-108.5
Reqd. RF Signal Level for S/N = + 1			+ 15.0 db
			- 93.5 dbm
RF Signal Level Margin	=	(-70.46 - 2.0) - (-93.5)	
	=	+21 db	

Thus, the RF signal margin is more than adequate to insure the required link performance.

SECTION 2. GLOBAL TO WAYSIDE COMMUNICATION LINK MESSAGE PROPOGATION DELAY ANALYSIS

A message originating at a Global Conter Center (GCC) to be sent to a Wayside Controller (WC) will have a dwell time calculated as follows:

$T_{total} = t_g + t_q + t_r + t_d$ = total dwell time
where t_g = time to generate/process at GCC.
t_p = propagation delay via FDDI data link
t_r = time to receive message at WC
t_d = time to decode and process message at WC.

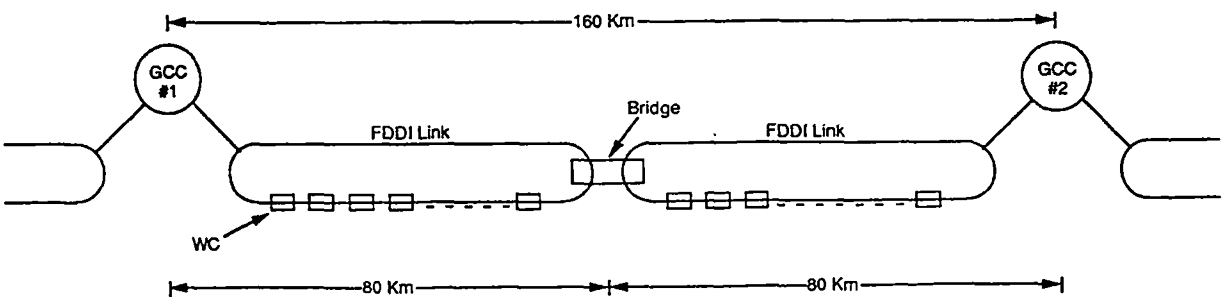
The message and FDDI link characteristics are defined as follows;

Message:
Length = 44 bytes = 484 bits
No. of WCs in FDDI link = 40 max.

FDDI data link:
Bit rate = 100MHz
Effective data rate = 85MB/sec
Message overhead = Preamble + CRC
= 112 bits + 48 bits = 160 bits
Processing Instructions/Station = 50
Length = 160Km (total loop)

Other Factors:
Speed of light = 3 x 10^8meter/sec
Computer clock rate = 25MHz

A simplified block diagram of the Global to Wayside FDDI data link can be shown as follows:



Calculating the total message dwell/delay time for a Global to Wayside transmission is as follows;

$t_g = t_{instructions} + t_{message\ bits}$

using 3 instructions per byte @ 25MHz rate

$$= 3(44)\frac{1}{25 \times 10^6} + 484\frac{1}{25 \times 10^6}$$

$$= 5.28\mu sec + 19.4\mu sec = 24.68\mu sec$$

$t_{process} = N\,(t_s + t_c)$
(FDDI)

where N = no. of stations/$_{link}$ = 40

t_s = time to generate message (preamble + data + CRC)

$$t_s = \frac{1}{100 \times 10^6}(160 + 484) = 6.44\mu sec$$

t_c = total link proprogation delay for 160Km

$$= \frac{160\ Km}{3 \times 10^8 m/sec} = 533\ \mu sec$$

then $t_p = 40\,(6.44 + 533.0) = 21.58$msec.

and $t_g \approx t_r \approx 24.68\mu sec$.

Calculating t_d;

Assume 100 computer instructions at 25MHz;

$$t_d = 100\frac{1}{25 \times 10^6} = 4.0\mu sec.$$

Calculating T_{total};

$T_{total} = t_g + t_p + t_r + t_d$

$= 24.68\mu sec + 21.58 \times 103\mu sec + 24.68\mu sec + 4.0\mu sec.$

≈ 21.63msec.

Note: T_{total} is calculated for the worst case condition of all forty (40) stations trying to access the link simultaneously.

A new probable T_{total} is that less than one half the stations would be accessing for the link, than

$$T_{total} = 24.68\mu sec + 20\ (6.44 + 533.0) + 24.68\mu sec + 4.0\mu sec$$

$$= 10.84 msec \approx 11 msec.$$

Thus, a message transmission from GCC #1 to GCC #2 will average approximately 22msec.

SECTION 3. MAGNEPLANE PASSENGER COMMUNICATIONS

On the back of each Magneplane seat, a flat electronics panel allows business travelers to plug in their portable lap top computers, place business calls through the Maglev communications center, and transmit or receive messages throughout the world. Passengers can flip a switch to reveal a flat panel six-inch LCD bright color television monitor that shows any one of twelve dual-language video programs. Another control gives passengers a microphone equipped headset access to any of thirty stereo audio channels including selected radio stations for their favorite listening. A built-in credit card reader provides for on board sales, instant accounting and enhanced revenue production for the Maglev system.

In addition, the Magneplane seatback units allow passengers to:

1. Access a numeric keypad for phone service and "video-shopping".
2. Display updated Magneplane arrival/departure and time-of-day.
3. Scan an on-screen video and programming guide.
4. Make automobile rental and/or hotel reservations.
5. Shop from a video catalog.
6. Display individual Magneplane station layouts.
7. Review Magneplane safety procedures.
8. Receive updated information and safety messages using text and/or audio/video.

3.1 PASSENGER COMMUNICATIONS IMPLEMENTATION

Each Magneplane vehicle has on-board dual-redundant communication control units (CCU). All Magneplane seatback units interface with the CCU. The CCU serves as the heart of the on-board passenger communications system with a built-in credit card reader and an automatic telephone switchboard. The CCU also provides the facility for recording on-board computer sales, an alpha-numeric keyboard, LCD video screen and printer.

A menu-driven software package guides the attendant through the system that allows the performance of such tasks as:

1. Control of the Magneplane vehicle audio/video system including variation of programs.
2. Displays messages from passengers.
3. Performs self-diagnosis (BITE).
4. Preview the video and audio entertainment system.
5. Lock-in and unlock external telephone links.
6. Print out credit card receipts when required.
7. Monitor inventory for on-board shopping.

Philips Airvision Ltd. makes seatback monitors that perform most of the on-board functions described above. They are currently under contract for installing their seatback monitors for several major commercial airlines. In addition to fast transportation, a Magneplane entertainment system as well as passenger telephone communications will help the Magneplane to compete with the automobile for inter-city travel. In order to provide external multiple telephone capability for passengers, the following is recommended. It should be noted that commercial airlines do not provide multiple service; however, this would be a desirable competitive feature for Magneplane passengers.

Time-division multiplex digital data transmission for voice and data through an RF link using slotted waveguide provides for the two-way passenger communications between the Magneplane vehicle and wayside control units. The wayside control units are then connected to the Globe Control Center through a dual redundant Fiber Distributed Data Interface (FDDI). The passenger voice information is distributed to the telephone networks at the corresponding Global Control Center.

The slotted waveguide installed along the inside of the Magneplane guideway provides broadband, contactless, 1.024 MB/s data transmission with very high reliability. This slotted waveguide technique has already been implemented in Germany. The frequency bandwidth between 2.3 and 2.5 GHz is used for local transmission between the Magneplane vehicle and wayside control units. The usable frequency band is between 2.2 and 2.6 GHz. The Magneplane vehicles are RF coupled to the slotted waveguide transmission line through a double antenna.

Some advantages of the slotted waveguide are:

1. No frequency planning dependent on FCC authorization or existing transmission services.
2. No erection of antenna masts.
3. Use of the same frequency band in the adjacent guideways.
4. Low transmitted RF power (~ 1m Watt).
5. The wide bandwidth provides for up to sixteen 64 KB/s voice channels or thirty-two 32 KB/s voice channels as well as room for future growth.
6. Low slotted waveguide power losses (15-20 dB/km).
7. Operate normally under bridges and inside tunnels.

Some of the slotted waveguide link capacity would be used as a backup to the wayside to on-board over-the-air land mobile RF link for operational voice and data. Operator determined override would be undertaken to invoke this feature. This would provide Maglev operator voice and data operation for an emergency situation.

System Concept Definition Report for the National Maglev Institute - Volume 8:Supplement D-I, Magneplane International, Inc, 1992 -