Refinements to a Parametric Entry, Descent, and Landing Design Tool for Mars Exploration

Matt Sorgenfrei[1] and Ed Chester[2]

[1]Department of Mechanical and Aerospace Engineering
University of California, Davis

[2]Space Research Centre
University of Leicester, UK

International Planetary Probe Workshop
20 June, 2012

Motivation

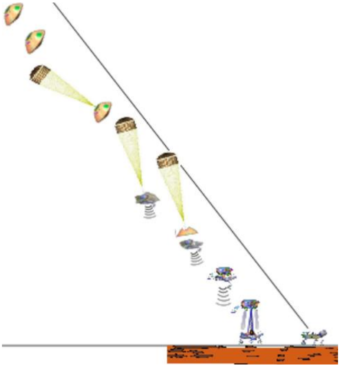


Figure: A representative EDL profile for operations on Mars

A Persistent Design Challenge

- Martian atmosphere is thick enough to create substantial heat but not sufficiently low terminal descent velocity
- Martian surface environment is very complex–rocks, craters, dust
- Interesting landing sites are at much higher elevations than previously explored
- Future missions require greater landed masses within substantially smaller landing ellipses than previously demonstrated

Not For the Faint-Hearted

Multi-Variable Design Optimization for Grown-Ups

- A wide range of design elements must be taken into consideration
    - Vehicle geometry, planetary models, aerodynamics, trajectory analysis, aerothermal, TPS, sizing
    - Each design element represents a computationally demanding study unto itself
- End-to-end EDL design requires leveraging large, data-intensive modules
    - Terrain elevation model (e.g. from MOLA), Mars-GRAM for atmosphere model, C_D,C_L look-up/generation tools, aerothermodynamic analysis, trajectory generation/analysis
- Even initial conditions for the design process are scary:
    - Altitude, velocity, flight path angle, azimuth angle, latitude, longitude, angle of attack, and bank angle

The Good News

Many Tools Exist for EDL Design

- Systems Analysis of Planetary Entry, Descent, and Landing (SAPE) from NASA Langley
- Planetary Entry Systems Synthesis Tool (PESST) from Georgia Tech
- Program to Optimize Simulated Trajectories (POST) from NASA Langley
- HyperProbe for aerothermodynamic analysis from San Jose State
- Planetary Mission Entry Vehicle (PMEV) Quick Reference Guide
- Just to name a few...

Some Challenges

How do we Integrate or Streamline These Tools?

- Is it possible to create a simple, user friendly package that supports efficient use of the vast array of detailed design tools?
- Does there exist a single, consistent set of parameters that can describe any design solution over a series of N flight phases?
- Is there merit to investigating the use of stochastic search tools, such as a genetic algorithm (GA)
    - Would result in exploration vs. exploitation of the design space
- Approach: Parametric Entry, Descent, and Landing Synthesis (**PEDALS**)

PEDALS Contributions

PEDALS is not an end-to-end design tool created to compete with other such systems currently available to the EDL market

What PEDALS Is

- An investigation into decision support algorithms with a particular emphasis on genetic algorithms
- A way to rapidly evaluate the space of EDL solutions
- A hybridization of stochastic search techniques and detailed deterministic models
    - Many models exist, goal is to leverage them efficiently
- A first step towards defining software interfaces that would allow rapid integration of 3rd party modules
- (Hopefully) a means to address the lack of access in Europe to ITAR-controlled software technology

PEDALS Contributions

PEDALS is not an end-to-end design tool created to compete with other such systems currently available to the EDL market

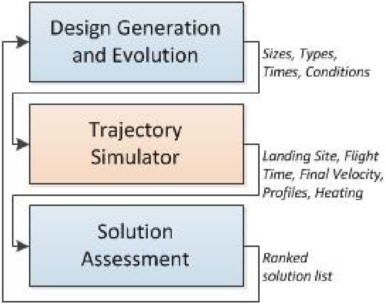
What PEDALS Is

- An investigation into decision support algorithms with a particular emphasis on genetic algorithms
- A way to rapidly evaluate the space of EDL solutions
- A hybridization of stochastic search techniques and detailed deterministic models
    - Many models exist, goal is to leverage them efficiently
- A first step towards defining software interfaces that would allow rapid integration of 3rd party modules
- (Hopefully) a means to address the lack of access in Europe to ITAR-controlled software technology

Overall Approach

Make Use of What Exists

- Optimization engine randomly creates and then evolves a population of design solutions
- This population of solutions is decoded into a usable form, passed to a trajectory tool
- Trajectory propagation requires access to terrain data, atmosphere data, etc.
- Output is passed to visualisation tool, evaluated per certain metrics


Design Generation and Evolution

Sizes, Types, Times, Conditions

Trajectory Simulator

Landing Site, Flight Time, Final Velocity, Profiles, Heating

Solution Assessment

Ranked solution list

Figure: Major components of the PEDALS system

Emphasis on Modularity

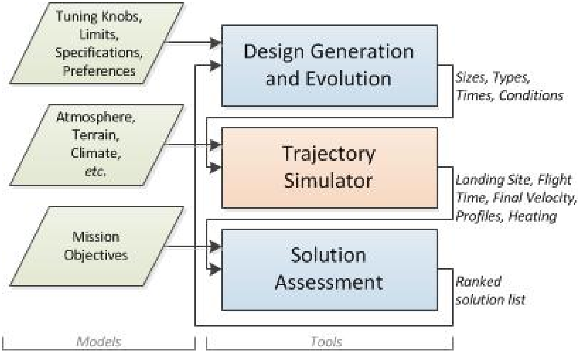


Figure: A wide range of models and databases can be used to support PEDALS

Emphasis on Modularity

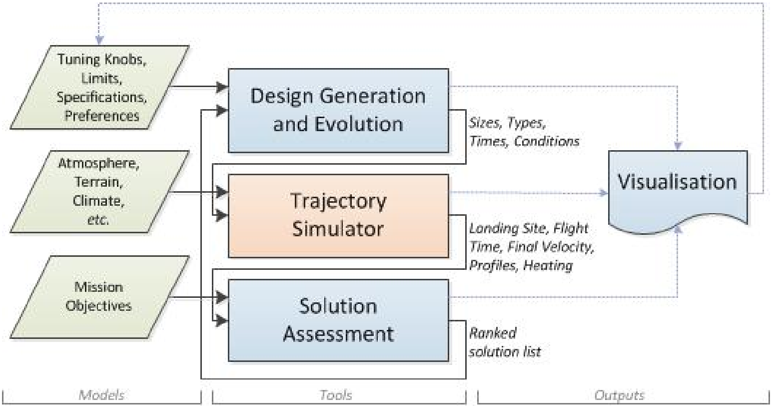


Figure: Ultimately, a stand-alone visualisation module could be used to provide a user-friendly interface

Some Possible Strengths

Flexibility Through Stochastic Search

- Randomized search of design space via GA does not guarantee entire space will be tested, however large area can be rapidly traversed
- For EDL, this could enable interesting design approaches
    - Fix entry interface and landed conditions, optimize flight path
    - Fix all initial conditions, optimize everything else
    - Optimize from fixed initial conditions then re-tune those conditions
- No limitation to number, type of tools that are implemented, possibly even in parallel
- Performance metrics can be tuned in a transparent, user friendly manner via the GA fitness function

Some Inherent Challenges

- How do we define discrete phases of flight? What happens at the interfaces between these flight phases?
- What is the best way to homegenize various initial condition standards and system assumptions?
- What is an elegant way to integrate the possibility of active control technologies?
- There are many ways for mass to change (TPS ablating, thrusters firing, heat shields dropping) that are connected to the phases of flight–integrating these discrete events with a continuous trajectory is tricky

Design Approach: Genetic Algorithm

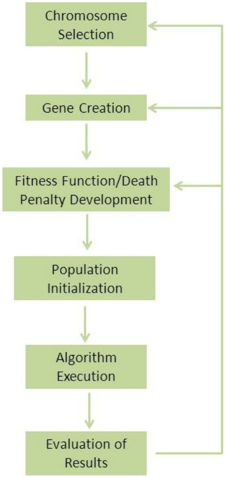
EDL Requires Multi-Variable Design Optimization

- Genetic Algorithms (GAs) have been widely used for complex optimization problems (Cage et al. 1994, Krishnakumar 1992)
- Design traits are encoded as genes, grouped onto chromosomes
- Allows designer to rapidly search a large portion of the design space in parallel–*n* designs evolved for *m* generations
- Driven by ‘natural selection’ and randomized evolutionary operators of crossover and mutation
- Natural selection dictated by a user-defined fitness function
- Design solutions that are deemed more fit as per the fitness function are more likely to ‘reproduce’ during algorithm execution

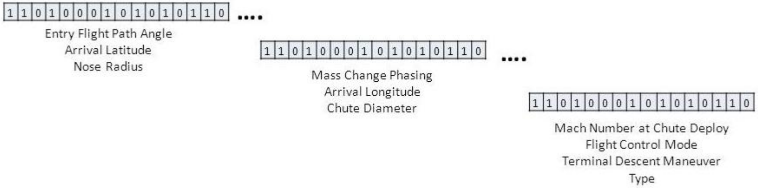
GA Implementation

Advantages to GA Approach

- Possibility of studying different design variable interdependencies through chromosome structure
  - Create one chromosome that tests different system initial conditions, another that trades on traditional design metrics
- Ability to adjust design variable resolution through gene allocation
- Flexibility of assessing multiple performance metrics via fitness function/death penalty

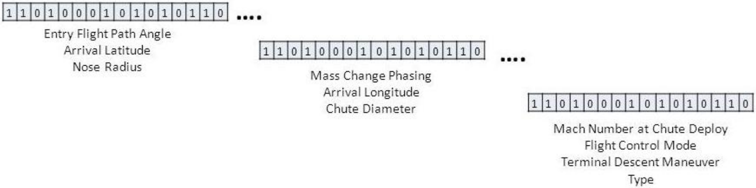

Chromosome Selection
Gene Creation
Fitness Function/Death Penalty Development
Population Initialization
Algorithm Execution
Evaluation of Results

Gene Structure



- Each design element can be encoded with a maximum of 16 bits
- Hybrid between discrete (e.g. flight phase transition) and continuous (e.g. nose radius) design variables
- Require design variable information for each phase of flight (currenly using entry plus 5 flight phases)–all stored as genes
- Crossover/mutation implemented as targeted bit flips, have to be careful about implications for hybrid structure

Gene Structure



- Each design element can be encoded with a maximum of 16 bits
- Hybrid between discrete (e.g. flight phase transition) and continuous (e.g. nose radius) design variables
- Require design variable information for each phase of flight (currenly using entry plus 5 flight phases)–all stored as genes
- Crossover/mutation implemented as targeted bit flips, have to be careful about implications for hybrid structure

Fitness Function Structure

The Driver for Natural Selection

- Objective is to maximize relevant figures of merit–for EDL this might included landed mass, velocity at impact, Euclidean distance from landing target, total heat flux
- Possible to combine function with a so-called death penalty that discards certain design solutions off-hand (e.g. if maximum deceleration limit is exceeded)
- When using multiple performance metrics user can include weight factors to achieve a certain balance

$$f = -(\alpha \cdot M + \epsilon \cdot V_f + R_f) \qquad (1)$$

Where M is the landed mass, V_f is the impact velocity, R_f is the distance from the desired landing location at impact, α and ϵ are user-tunable weight factors

Fitness Function Structure

The Driver for Natural Selection

- Objective is to maximize relevant figures of merit–for EDL this might included landed mass, velocity at impact, Euclidean distance from landing target, total heat flux
- Possible to combine function with a so-called death penalty that discards certain design solutions off-hand (e.g. if maximum deceleration limit is exceeded)
- When using multiple performance metrics user can include weight factors to achieve a certain balance

$$f = -(\alpha \cdot M + \epsilon \cdot V_f + R_f) \tag{1}$$

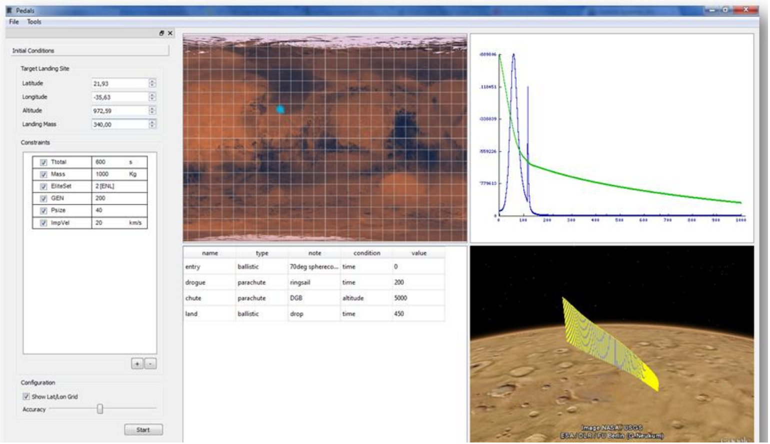
Where M is the landed mass, V_f is the impact velocity, R_f is the distance from the desired landing location at impact, α and ϵ are user-tunable weight factors

Implementation Considerations

The Problem Within the Problem

- GA parameters–population size, number of generations, crossover and mutation rates all impact performance of the algorithm
- Past research (Grefenstette 1986) provides some suggestions as to viable design combinations
- EDL is a hard, multi-disciplinary problem–possible that past research is not valid for this work
- Inclusion of multiple heterogeneous components (trajectory simulators, aerothermodynamic models, spacecraft geometry assessment) could be both a benefit and a liability for algorithm execution

Integration with Software Front-End



Initial Conditions

Target Landing Site

Latitude	21,93
Longitude	-35,63
Altitude	972,59
Landing Mass	340,00

Constraints

☑	Ttotal	600	s
☑	Mass	1000	Kg
☑	EliteSet	2 [ENL]	
☑	GEN	200	
☑	Psize	40	
☑	ImpVel	20	km/s

Configuration

☑ Show Lat/Lon Grid

Accuracy

name	type	note	condition	value
entry	ballistic	70deg sphereco...	time	0
drogue	parachute	ringsail	time	200
chute	parachute	DGB	altitude	5000
land	ballistic	drop	time	450

Figure: PEDALS user interface and flight visualisation engine

Conclusions

- A major goal of PEDALS is to create a modular system that is comprised (mostly) of open interfaces
- The whole system will only function as a result of the work being done by experts in this room
- PEDALS v0.3 would love inputs in the form of data/models/tools
- Any and all suggestions for improvement are more than welcome

References I

1. Braun, R.D., and Manning, R.M. "Mars Exploration Entry, Descent, and Landing Challenges" Proc. of IEEE Aerospace Conference, Big Sky, MT, 2006.
2. Steltzner, A., et al., "Mars Science Laboratory Entry, Descent, and Landing System", Proc. of IEEE Aerospace Conference, Big Sky, MT, 2006.
3. Munk, M., Overview of the NASA Entry, Descent and Landing Systems Analysis (EDL-SA), IPPW-7, Barcelona, Spain, 2010.
4. Raiszadeh, B., Queen, E.M., and Hotchko, N.J., Validation of Multibody Program to Optimize Simulated Trajectories II (POST II) Parachute Simulation with Interacting Forces, Hampton, Virginia, NASA TM-2002-211634, 2002.
5. Samareh, J.A., "A Multidisciplinary Tool for Systems Analysis of Planetary Entry, Descent, and Landing (SAPE)," NASA TM-2009-215950
6. Otero, R.E. and Braun, R.D., "The Planetary Entry Systems Synthesis Tool: A Conceptual Design and Analysis Tool for EDL Systems," Proc. of IEEE Aerospace Conference, Big Sky, MT, 2010.
7. Papadopoulos, P. and Subrahmanyam, P., "Web-based Computational Investigation of Aerothermodynamics of Atmospheric Entry Vehicles," Journal of Spacecraft and Rockets, Vol. 43, No. 6, pp. 1184-1190, 2006.

References II

8. Alemany, K., Wells, G., Theisinger, J., Clark, I., and Braun, R., "Mars Entry, Descent, and Landing Parametric Sizing and Design Space Visualization Trades," Proc. of AIAA/AAS Astrodynamics Specialist Conference, Keystone, CO, 2006.
9. Bayley, D., Hartfield, R., Burkhalter, J., and Jenkins, R., "Design Optimization of a Space Launch Vehicle Using a Genetic Algorithm," Journal of Spacecraft and Rockets, Vol. 45, No. 4, pp. 733 - 740, 2008.
10. Cage, P.G., Kroo, I.M., and Braun, R.D., "Interplanetary Trajectory Optimization Using a Genetic Algorithm", Proc. of AIAA Astrodynamics Conf., Scottsdale, AZ, 1994.
11. Krishnakumar, K., and Goldberg, D.E., "Control System Optimization Using Genetic Algorithms", Journal of Guidance, Control, and Dynamics, Vol. 15, No. 3, 1992, pp. 735 - 740.
12. Grefenstette, J., "Optimization of Control Parameters for Genetic Algorithms," IEEE Trans. on Systems, Man, and Cybernetics, Vol. 16, No. 1, pp. 122-128, 1986.
13. Sorgenfrei, M., and Joshi, S., "Reconfigurable Spacecraft Controller Design via Location-Scheduled Control", Journal of Guidance, Control, and Dynamics, Vol. 34, No. 5, pp. 1598-1600, 2011