QUICK-LAUNCH SPACETUG MATE PROJECT
(Homeworks 5, 6, and 7)

The customers for the SpaceTug project liked the result, but can see an area where the tradespace needs to be expanded. The customers wish to see the concept expanded to include various storage and launch options. Storage options might include none (build vehicle to need), ground "cold" (in a box), ground "hot" (fueled and powered, and perhaps mated to launch vehicle), orbit cold (sleeping, needing to be powered up and tested out), orbit hot (systems on and tended by ground operators on a regular basis), and perhaps others. Launch options might include conventional (get in line for an un-prioritized launch), prioritized launch, "cold" dedicated boosters (waiting unfueled at manufacturing facilities or near the pad), "hot" dedicated boosters (waiting launch-ready on or near the pad), and perhaps others. Both current and reasonable future capabilities should be considered. In particular, the possibility of "quick launch" capability and the added utility that could be gained by having it is of interest. Google "DARPA Falcon" for some current ideas about quick launch capability.

Part I (Homework 5)

The SpaceTug concept started with an assumption that the tug would be based in a parking orbit. Thus its response time would be governed entirely by the thrust of its propulsion system. This assumption lead to the binary model of the response time attribute – it was either fast (1) or slow (0). This assumption no longer holds, so a proper response time attribute, with units, a range, and a single attribute utility curve is needed.

We will define response time as the *time from the perception of a need for tug services to the arrival of a tug in an orbit near the target vehicle*. We will leave open what the tug is going to do when it gets there, so will not worry (for now) about times for docking, tugging, repair, etc; we will also leave somewhat open the definition of "near orbit," as it rather depends on what the tug is going to do next. We could conceive of uses for a tug that would require response time anywhere from hours (astronaut rescue) to years (disposal of dead vehicles in useful orbits).

a) Fix a unit and a range for the response time attribute and confirm them with the proxy user. Prof. Hastings will be the proxy user – he is, in fact, an ideal one, being aware of the needs of the real users without being invested in either their individual needs or their pet solutions.
b) Create a lottery-equivalent interview, and interview Prof. Hastings to create the single attribute utility curve for response time.
c) Propose design vector element or elements to capture the expansion of the tradespace outlined in the introductory paragraph above. Confirm with Prof. Hastings that it covers the concept space of interest adequately.

Part II (Homework 6)

The larger design vector, and the desire to more precisely specify response time, requires us to update the simple SpaceTug model. The goal is not to greatly increase its complexity or level of detail. The need is to capture, at about the same (low) level of detail, the new issues raised by the expanded scope being considered. Use the SSPARC book sections for Weeks 3 and 5 (in particular, the Spacetug examples) as templates for this work. Do not be afraid to make the models parametric—put the parameters in the constants vector and explore their effects next week.

a) Use a QFD matrix, some brainstorming and discussion, and if necessary some expert opinion (Prof. Hastings, Dr. McManus, and Adam Ross are all good resources for this) to decide how the new design vector will affect the Spacetug attributes.
b) Decide the key issues that need to be modeled. Identify the assumptions that you need to make to keep these models tractable. If possible, quantify these assumptions, and add the quantities to an expanded constants vector. Identify the modeling that needs to be done, and any new intermediate variables that need to be calculated to link the models.
c) Create a new SpaceTug model. Write out the mathematics. Implement them on any tool you find convenient. The old spacetug sheet enumerated the entire tradespace on lines of an excel spreadsheet; you may still be able to do this, or you may need to switch to an algorithmic model (C, VB, or Matlab).
d) Verify the model by running test cases. The modeling should be simple enough you can do at least a few cases by hand, and the model should have the existing tradespace as a sub-set, so you have things to test against. Also, run extreme cases to check that you get answers in the right extreme directions.

Part III (Homework 7)

Explore the tradespace you have created! The point here is to develop (new) knowledge. There are no right answers. If you seem to be getting very wrong answers check the modeling assumptions and/or coding. At a minimum:

a) Generate a big-picture plot of the tradespace. Identify the Pareto front and the key trades or issues along it.
b) Identify the impacts of the various design vector elements on the tradespace by color-coding big-picture plots or creating other plots of the tradespace.
c) Parametrically explore the tradespace for various users, considering the same types of users considered in the original SpaceTug study.
d) Study the sensitivity of the tradespace to various quantified assumptions by varying the corresponding constants vector elements up and down and documenting the effects on the tradespace.
e) Make recommendations to the user on “best” architecture(s), key technologies that need to be pursued, and key sensitivities and/or assumptions that might affect or change your recommendations.