Tool-Supported Mechatronic System Design

R A Hyde & J Wendlandt
The MathWorks, Inc. 3 Apple Hill Drive, Natick, MA 01760-2098
rick.hyde@mathworks.com

The design of quadruped trotting robot is considered as an example of a complex mechatronic system. This is used to illustrate some of the challenges in developing mechatronic systems, and how dynamic models can be used to communicate between design groups and with suppliers. The importance of having a common simulation platform for design activities is argued. The platform must also support model refinement, validation and abstraction. This enables engineers to work with models with appropriate levels of fidelity, ranging from system-level for design trade-off studies, to implementation-level for detailed component design.

I. INTRODUCTION

Mechatronic system design requires consideration of dynamics and control aspects when making fundamental design choices. This, combined with the increasing number of available actuator and sensor technologies, can provide some unique challenges. An illustrative example is development of a flapping wing micro-UAV [1,2] where minimizing overall system weight is just as critical as producing the necessary complex wing motion. Finding a good or best design requires assessments of a wide range of technologies, plus some subsequent optimization of the particular solution chosen. Designing a bespoke actuation solution for an industrial automation application may also require optimization to reduce cost whilst delivering to a performance specification.

New actuation solutions already find application in a wide range of industries. For example, a novel electro-hydrostatic actuator is being used on the Airbus A380 to reduce weight [3]. In robotics, high torque brushless motors now provide a viable alternative to hydraulics, and pneumatics has shown itself to be a technology suitable for applications requiring precise control [4]. Smart materials [5] also provide many new alternatives, but can also bring manufacturing challenges as they may need to be fabricated as part of the main system.

As well as a range of off-the-shelf actuation solutions, the designer may also consider hybrid solutions where two actuators are combined to produce a solution with the best properties from both. An example is the use of hybrid drives in the automotive industry [6], or combining a pneumatic actuator with a series piezoelectric motor to give both large stroke and fine control [7].

System design at the concept level is traditionally supported by methods from engineering design, including the function-means approach to break down requirements, morphological charts to explore solution combinations, and weighted objectives to compare solutions [8]. For more complex mechatronic systems, this type of analysis may need underpinning with system-level simulation models. Here *system-level model* implies only a few parameters to tune, and fast simulation to allow design trade-offs to be understood.

One approach when designing a new system is to make incremental changes to a previous design. However, with this evolutionary method the previous simulation model is likely to be an *implementation model*. Here *implementation model* implies that it is detailed, possibly integrating multiple models from suppliers with many hundreds of parameters. It is hard to make design trade-offs and even harder to use the models to consider novel actuation solutions that are potentially disruptive to some aspects of the design.

In this paper it is shown how a common simulation platform can be used to use both system-level and implementation-level models to best effect. System-level models can be used early on to understand design trade-offs, and then later refined into implementation models. A continual process of validation is required to ensure that both sets of models remain consistent with each other. This may also require ways to abstract behavior from an implementation model back up to a system-level model. The common simulation platform used is Simulink®, and the application example is a simulated quadruped robot. The example is suitably complex to tease out the tool requirements that would be seen across a range of different applications.

The overall design process is outlined in Section II, a key feature of which is the use of models to communicate between the different design activities. A list of the key requirements for a tool-supported design process is then derived from the process. Section III then illustrates these requirements with application to the quadruped robot. Section IV then reviews what information is passed between design stages and models, and identifies the opportunity to automate to support design iterations. Finally conclusions are drawn in Section V.

II. DESIGN PROCESS

A. Requirements

To illustrate this paper, the design of a four-legged running robot is considered. Possible applications include search and rescue, transportation to inaccessible locations and to support research into animal terrestrial motion. The Boston Dynamics Big Dog robot [9] is an example of similar application. It uses hydraulic limb actuation plus a gasoline-driven generator to provide electrical and hydraulic power. In this paper it is assumed that stealth, high speed and reduced range requirements result in the choice of electric actuation plus a battery power source.

To manage complexity, single-plane motion is considered. This is a realistic constraint early on in the design as this is sufficient to capture the energy requirements and predict range. The running gait is bio-inspired, and is akin to the trotting gait of a horse. To make this possible, the limbs must behave like linear springs [10]. A galloping gait could also have been explored as a way to provide more speed, but with the cost of reduced efficiency [11]. Overall robot weight is set at 50 kilograms, including payload, ensuring that two people can lift it. Leg length is set to 1 meter, and is chosen so that the robot can handle typical terrains.

B. An Example Design Process

Fig. 1 shows a schematic top-level view of a design process for the locomotion system, focusing primarily on the hip actuator. It has been assumed that the hip actuator is designed by a component supplier to meet the prime contractor's specification. A number of models are used to support the design from conceptual design to implementation. Dotted lines connecting the models imply that there is abstraction of data from one model to the next, and solid lines imply that the model is used as-is as part of larger model.

Models provide a more reliable way than paper documents to communicate information between design tasks and organizations. Here Model 3 is used to communicate a system-level specification to the supplier, and the supplier's detailed Model 4 is validated against this. Model 3 can then be used as part of the overall System Model to support simulation and trade-off studies. The System Model is also used to test the hip actuator hardware by generating real-time C code that is used to exercise the actuator in real time.

At the conceptual design stage, top-level requirements are broken down using design methods. The locomotion system can be broken down into five sub-systems, namely the energy source, energy converter, actuation system, gearing and leg. Solutions to each of these categories can then be broken down using the function-means method [8]. For example, the actuation requirement might be broken down into hydraulic, pneumatic, electric and chemical actuation types. Each of these can then be explored in combination with the other categories using a morphological chart. For this paper, it is assumed that this process results in the selection of a high-power battery, brushless permanent magnet motor, mechanical gearbox and a leg with a folding knee joint.

At the conceptual stage, simple dynamic models may be required to support the design choices made. This is particularly true for mechatronic systems where different actuation solutions provide widely different dynamics, as well has having a range of torque per unit volume or torque per unit mass properties. It will be shown for the robot example that the actuator must be able to provide high torque levels that can be modulated over a few tenths of a second, ruling out pneumatic actuation.

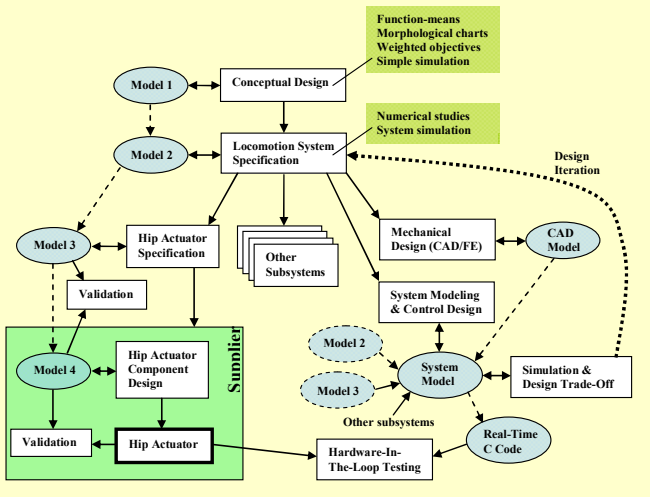

Fig. 1. Locomotion system design process. Parts of the process related to the hip actuator are shown, and how models are used to communicate between steps.

C. Tool Requirements

Figure 1 suggests that a common simulation platform should support the following activities:

R1: System-level modeling. System-level models exhibit emergent behavior without detailed simulation of underlying details. They are used to make design trade-offs, and also communicate requirements to a supplier.

R2: Multi-domain modeling to support electric, hydraulic, pneumatic and any other physical domains. This requires fundamental blocks to support system-level studies, and more detailed blocks to support component design.

R3: Implementation-level or component-level modeling. These models include all relevant implementation detail so that they can be reliably used for detailed design and validation of abstracted system-level models.

R4: Control system design. The models must be linearizable, and capture any dynamics that contribute gain and phase around open-loop cross over frequencies.

R5: Model abstraction. This supports re-use of existing models from previous designs, and may also be useful to extract a system-level model from a supplier implementation model.

R6: Design optimization. Once the system architecture has been chosen with the aid of system-level modeling, it can be parameterized and then optimized to user requirements. Optimization relies on simple abstracted models that have a few parameters to tune, and that run fast.

R7: Data management across models. For example, the abstraction of total subsystem weight from a supplier model to a top-level system model.

These tool requirements are now illustrated in the context of the running quadruped robot and the Simulink® product family. This is with the exception of model abstraction (R5) which will be addressed in separate paper on ways to abstract from a supplier implementation model to a system-level model.

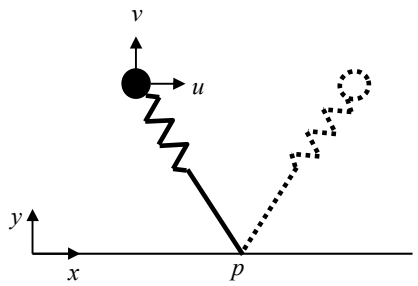

Fig. 2. Mass-spring hopping gait model.

III. QUADRUPED RUNNING ROBOT

A. *Model 1 – Conceptual Design (R1)*

In order to understand the trotting gait, the inverted pendulum and spring-mass model proposed in [10] to model animal terrestrial motion is used. Based on Figure 2 the equations of motion are:

$$m\dot{u} = k(L - L_0)\sin\theta \quad (1)$$

$$m\dot{v} = k(L - L_0)\cos\theta \quad (2)$$

$$L = \sqrt{(x-p)^2 + y^2} \quad (3)$$

where L_0 is the unloaded leg length.

As these are ordinary differential equations (ODEs), they easily implemented in Simulink using integrators for the motion states as shown in Figure 3. Mass is set to 25kg i.e. half of the total system weight. Leg length is also assumed fixed at 1m, and desired running speed is 2m/s. This leaves the spring stiffness and the gait shape to be selected. A MATLAB ® m-file is used to exercise the model over a range of spring stiffnesses, and plot the resulting gait characteristics. Figure 4 shows a trade-off plot of % of gait time spent in contact with the ground as a function of leg stiffness. Plots are shown for different spring compression levels *y0* at the mid-stance point.

A stiffness of 0.77 KN/m and a 50% of gait spent in stance is chosen. This relatively long stance time provides longer for the hip motor to provide propulsive torque to maintain forward speed. The selected maximum spring compression of 0.9 times unloaded length is a trade-off between providing sufficient clearance for ground obstacles and mechanical energy losses. These choices result in a gait period T of 0.75

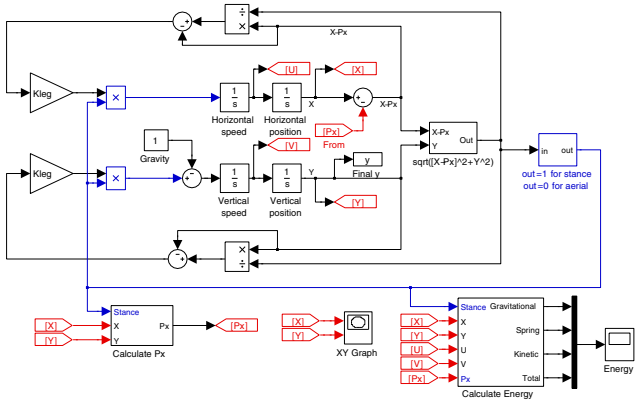

Fig. 3. Simulink implementation of the mass-spring hopping gait model.

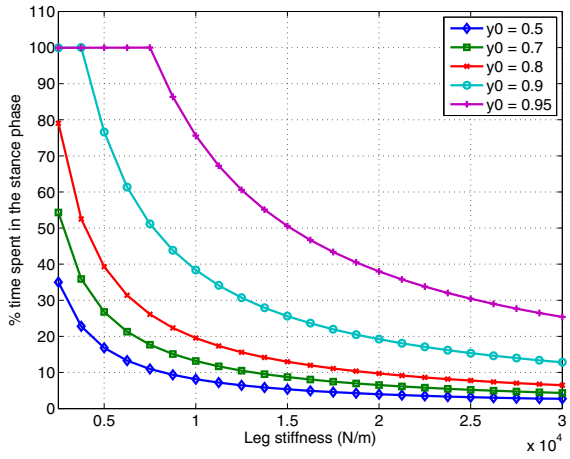

Fig. 4. Gait analysis results obtained from running the model in Fig. 3 over a range of leg stiffness values and initial mid-stance heights.

seconds. In reaching a decision on spring stiffness and gait shape, other scenarios could also be taken into account, including running up/down inclines and running on soft ground. This model is an example of a system-level model as denoted by requirement R1 in Section IIC.

Model 2 – Hip & Knee Torque Requirements (R1)

Having determined spring stiffness and the gait profile, a second simple model is used to determine the actuator torques required to swing the leg forward ready for the next stride. Figure 5 shows the model which consists of a hip joint, upper-leg, knee joint and lower leg. The leg components are modeled as equal-length rods with uniform density, and each weighing 5% of the 25kg weight. The leg has one other design parameter which is the no-load knee angle which is set to 130 degrees. This angle and the overall leg stiffness of 7.7KN/m can be used to determine the required rotational spring value at the knee.

The model is implemented in SimMechanics™, The MathWorks multi-body simulation package. SimMechanics works seamlessly with Simulink®, and uses a projection method to provide efficient simulation of 3-D mechanical systems with minimum set of dynamic equations. The model is run in reverse-kinematics mode whereby the motion is specified, and resulting torques are logged.

The desired motion during stance is taken from Model 1, and the hip and knee angles during the aerial phase is assumed sinusoidal. As alternate left and right legs are used during the trot, the time available to swing the leg forward

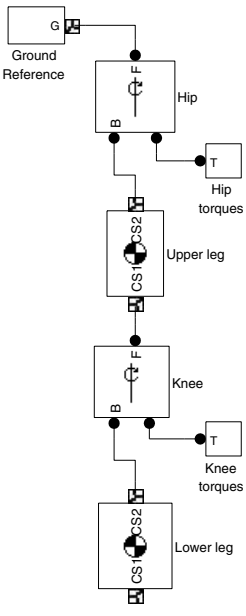

Fig. 5. Model 2 - simple leg modeled in SimMechanics™

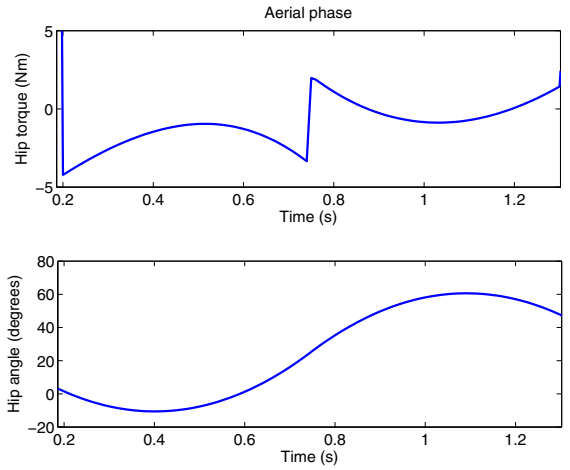

Fig. 6. Torque and speed requirements from running the model in Fig. 5.

is 3T/2. The hip sinusoid frequency, phase and amplitude parameters are picked to match hip speed and angle at end of previous stance phase, and the beginning of the next stance phase. The knee angle sinusoid profile is picked to retract the knee angle sufficiently during the aerial phase. Fig. 6 shows the torque and speed profiles at the hip obtained from running Model 2.

Model 2 is an example of a behavioral model as denoted by requirement R1 in Section IIC.

Model 3 – Hip Actuator Specification (R1,R2,R4)

Given the hip torque profile obtained from Model 3, it is now possible to develop a top-level specification for the hip actuator. This specification takes the form of a model that is used for control design, incorporation into a complete system model, and also to communicate requirements to the hip actuator supplier. Key requirements of this model are therefore that it includes relevant dynamics for control design, and fast simulation to support system-level design.

The actuator consists of mechanical and electrical parts, and the underlying equations are a combination of ODEs and Differential Algebraic Equations (DAEs). The Mathworks physical modeling tool is Simscape™, and provides basic blocks for electrical, mechanical, hydraulic and thermal systems. SimElectronics™ provides additional mechatronic capability. Figure 7 shows a Simscape/SimElectronics model of the DC power supply, motor, gearbox, simple controller and inertial load. The Simscape model is used to pick a gearbox ratio and a suitable motor. It is also used to design a motor speed and angle controller that tracks the motion demands produced by running Model 2.

The Servomotor and Driver block provides a system-level model of a controlled brushless permanent magnet motor. Electrical power drawn from the DC supply matches the mechanical power delivered plus resistive i^2R losses. The internal torque controller has an associated parameterized lag, and limits output torque to be within a torque envelope that the user specifies as a parameter. These parameters match the characteristics typically provided in manufacturer datasheets.

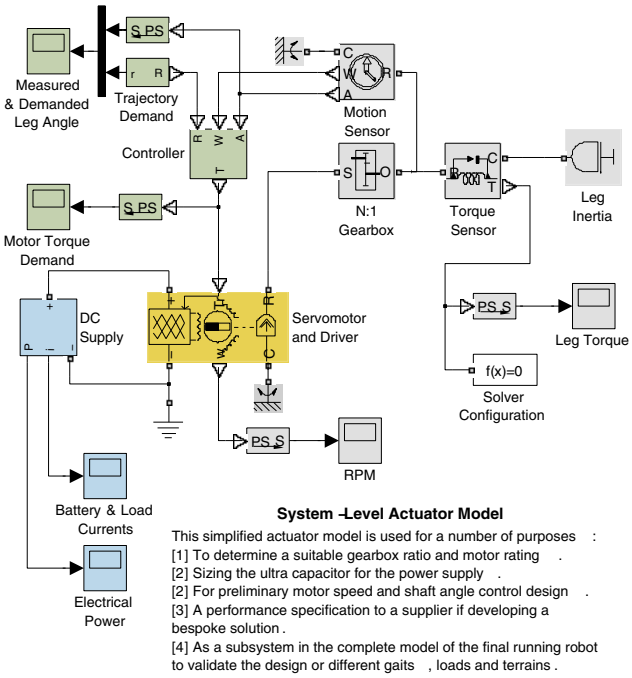

Fig. 7. Model 3 - Simscape™ implementation of the hip actuator.

Using the Simscape model, the gear ratio is set to 80 resulting in a maximum motor speed of 2700 rpm and motor torque of 0.08Nm for this gait. The selected motor has a maximum torque of 0.13 Nm falling to 0.1 Nm at 4000 rpm, an inertia of 0.05 kg cm^2 and a mass of 0.83 kg. This mass value can now be used to review & revise assumptions made in the locomotion system specification if necessary.

The hip angle controller is designed to have an open-loop bandwidth of 50 rad/s. An upper constraint on the bandwidth is set by the servomotor torque time constant which at 0.01 seconds adds 27 degrees phase lag at 50 rad/s. Simulink® Control Design ™ can be used to visualize the open-loop Bode plot and select controller gains that ensure a good performance and robustness trade-off. The controller feedback signal is a composite of both angle and angular rate information, this following the structure proposed used in [12] to achieve good position tracking at the same time as good phase margins.

Model 3 is an example of meeting requirements R1, R2 and R4 defined in Section 2.3. It is a simple system-level (R1) with a few tunable parameters, plus it is multi-domain (R2) taking account of electrical and mechanical energy

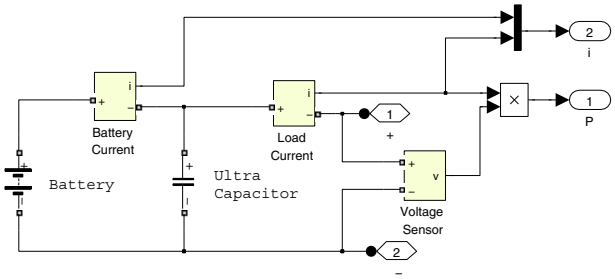

Fig. 8. System-level DC supply model used to size the ultra capacitor.

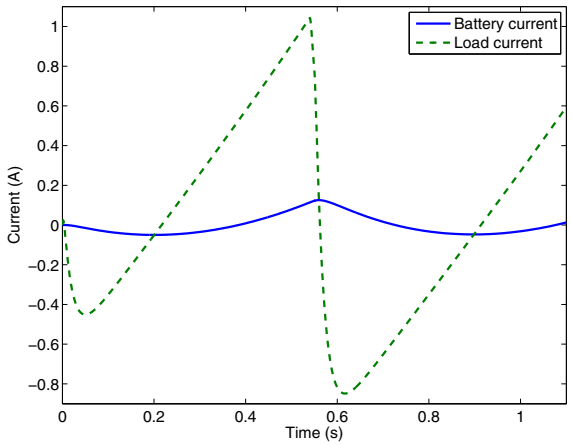

Fig. 9. Battery and load currents during the aerial phase.

interchange. Moreover, it is linearizable (R4) to support control system design.

B. Model 3 – Power Supply Requirements (R1,R2)

Model 3 can also be used to support the power supply design. The peak current drawn by the motor is 1 amp. This value could be used to size the battery, but this might result in an oversized battery. A battery must be chosen to both deliver both the peak current, and the total ampere-hours required. Also, in this application power is supplied back to the battery during leg deceleration, and not all battery types can be charged at the same rate as they can discharge.

Figure 8 shows the Simscape implementation of the power supply. An ultracapacitor is used to smooth the current drawn from the battery. Ultracapacitors are suited to very high charge and discharge rates, and also give very good energy storage per unit mass. Figure 9 shows the resulting battery and load currents when the ultra capacitor has an effective capacitance of 1 Farad. A typical mass per Farad is 0.15 kg which must be included as part of the total battery weight budget.

C. Model 4 – Detailed Component Design (R3)

The actuator supplier will typically use a detailed model of the servomotor and its controller. An ideal platform for this is the Simulink® add-on SimPowerSystems™. The model would include detail such as the PWM switching to control the motor current. By means of a suitable test harness, this model would be run back-to-back with Model 3 along with the loads defined in Model 3 to validate that the desired torque tracking is achieved. This model is an example meeting requirement R3 defined in Section IIC.

D. Model 5 - System Model (R1,R2)

The complete system model is shown in Figure 10. The system model integrates subsystem models from multiple sources, including the hip actuator model (Model 3). Simulink® provides two mechanisms for model-component reuse, namely Model Reference blocks and Simulink libraries. These enable a component defined in one place to instantiated in multiple places, either within one model, or

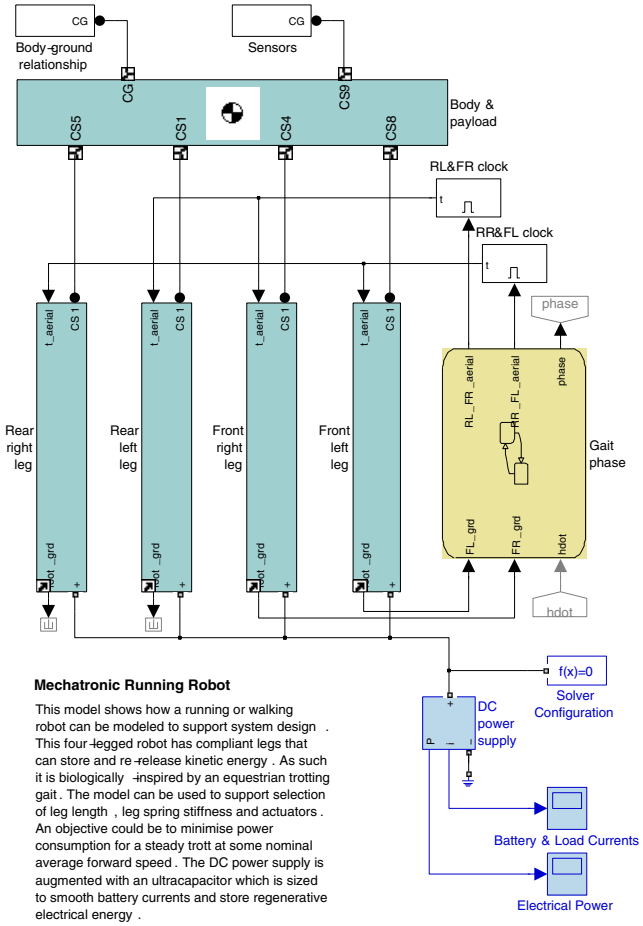

Fig. 10. Model 5 – Complete system model.

across multiple models. Here, the hip actuator model is instantiated four times in the complete system model. Building the system model also requires abstraction of data from the CAD model as illustrated by a dotted arrow in Figure 1. As the CAD and FE models of the leg are refined, the mass-distribution assumptions for the upper and lower leg are iteratively improved. The MathWorks provides link tools that allow the importing of SolidWorks and Pro/ENGINEER models into SimMechanics™.

IV. AUTOMATING FOR RE-DESIGN

Table 1 summarizes the models used to support the design and the data passed between the models. Design by its very nature is iterative, but the associated delay can result in missing a market opportunity. Moreover, fast iteration also fosters better understanding of the inherent design trade-offs and may admit a better design.

Using models as a way to communicate between design stages and/or design groups can have a beneficial impact on reducing this iteration delay. Table 1 shows that the complete path from top-level requirements all the way to full system model can be automated. This in turn means that the system engineer can vary these top-level design parameters and

lower-level assumptions to determine the impact on final performance.

The design example provides strong motivation for Requirement R7 Section IIC for data management across models. Given automated data flow between models, Table I shows that the impact on range of leg length, running speed, aerial-to-stance ratio, knee angle, leg retraction angle can all be explored in an efficient way. Optimization based approaches can also be used to find a best solution. This was identified as Requirement R6 in Section IIC. Two tool options that support this are the Optimization Toolbox™, and Simulink® Response Optimizer™. An optimization-based design approach is proposed in [13] for a robot application.

V. CONCLUSIONS

This paper has explored the benefits of tool-supported mechatronic system design. With the aid of a robot design example, the main modeling tasks have been identified and then illustrated using the MathWorks toolset. Central to the approach is the use of models to help communicate between top-level and low-level design tasks. This requires an integrated modeling environment, such as that provided by the MATLAB language and Simulink simulation platform.

It has been suggested that mechatronic systems provide additional design challenges, and that use of system modeling is a practical way to support and inform conventional mechanical system design methods such as function-means based analysis and CAD tools. Providing tools that do not require the designer to work extensively at the equation level allows design effort to be expended at the system level and creates a better understanding of the design trade-offs.

It has been suggested that this example admits automation of the design decisions based on the use of each of the simulation models. Automation allows the designer to produce top-level trade-off plots of system requirements against system parameter choices, such as the effect of leg length or payload mass on range.

The scope for using optimization in conjunction with system-level models has been identified. Optimization allows identified best candidate solutions to be refined against top-level requirements such as range.

REFERENCES

[1] Conn, A.T., Burgess, S.C. and Hyde, R.A. "Development of a novel flapping mechanism with adjustable wing kinematics for micro air vehicles", 3rd International Design & Nature Conference, May 2006.

[2] Conn, A.T., Burgess, S.C., Hyde, R.A. and Chung Seng Ling, "From Natural Flyers to the Mechanical Realization of a Flapping Wing Micro Air Vehicle", IEEE International Conference on Robotics and Biomimetics, 2006, pp. 439-444.

[3] Richardeau, F. Mavier, J. Piquet, H. and Gateau, G., "Fault-tolerant inverter for on-board aircraft EHA", European Conference on Power Electronics and Applications, 2-5 Sept 2007, pp1-9.

[4] Colbrunn, R.W. Nelson, G.M. Quinn, R.D, "Design and control of a robotic leg with braided pneumaticactuators", Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 29 Oct – 3rd Nov 2001.

[5] Huber, J.E., Fleck, N.A. and Ashby, M.F., "The selection of mechanical actuators based on performance indices," 1997, Proceedings of the Royal Society [Mathematical, Physical and Engineering Sciences], Vol. 453, No. 1965, pp. 2185-2205.

[6] Saski, S., "Toyota's newly developed hybrid powertrain", Proceedings of the 10th International Symposium on Power Semiconductor Devices and ICs, Kyoto, Japan, 3-6 Jun, 1998.

[7] Yung-Tien Liu and Higuchi, T. , "Precision position device utilizing impact force of combined piezo-pneumatic actuator", IEEE/ASME Transactions on Mechatronics, Vol. 6, Issue 4., Dec 2001, pp467-473.

[8] Pahl, G. and Beitz, W. "Engineering Design: A Systematic Approach", 1995, (translated by Wallace, K.), Springer-Verlag, ISBN: 3540199179.

[9] Boston Dynamics' *Big Dog* http://www.bdi.com/content/sec.php?section=BigDog

[10] Blickhan, R. "The spring-mass model for running and hopping", 1989, J. Biomech. 22, 1217–1227.

[11] McNeill Alexander, R."Principles of animals locomotion", 2003, Princetown University Press. Princetown NJ.

[12] R.A. Hyde and K. Glover "The application of scheduled H-infinity controllers to VSTOL aircraft". IEEE Transactions on Automatic Control, July 1993. Volume 38, Number 7.

[13] Krasny, D.P. and Orin, D.E. "Generating High-Speed Dynamic Running Gaits in a Quadruped Robot Using an Evolutionary Search", IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics, Vol. 34, No. 4, August 2004.

TABLE I
SUMMARY OF DATA FLOW BETWEEN MODELS

	Inputs	Outputs	Automation steps
Model 1	Design parameters: From requirements analysis, running speed u=2m/s, total mass M=50kg, leg length L=1m, Range R	Leg stiffness K Gait period T Stance ratio S Stance height y0	MATLAB script takes design parameters u, L and M, and produces trade-off plots. To automate, user also specifies stance ratio and max spring compression which in turn dictates leg stiffness.
Model 2	Inputs from Model 1: K, T, S, y0 & trajectory. Design parameters: Unloaded knee angle & leg retraction angle. Assumptions: Leg mass as % of M, sinusoidal leg swing	Hip torque profile Hip speed profile Knee torque profile Knee speed profile	MATLAB script reads data stored from running Model 1. Script also has user-defined unloaded knee angle and degree of leg retraction required. Script runs the model & stores hip and knee torque & speed profiles.
Model 3	Inputs from Model 2: Hip torque & speed profiles. Design parameters: Gearbox ratio, motor torque-speed profile, ultra capacitor sizing, hip angle controller bandwidth.	Gearbox ratio Motor torque-speed characteristics Controller gains Ultra capacitor size	MATLAB script reads hip torque and speed profiles stored from running Model 2. Tabulate & fit simple functions that define torque/kg and inertia. From this the gear ratio can be determined & hence torque requirement. Fix controller bandwidth & hence determine the necessary controller gains.
System Model	Re-use four copies of the leg model in Model 2. Re-use four copies of the hip actuator & controller in Model 3. Use target motion profiles from Model 2	Use the model to fine-tune ultra capacitor sizing, select battery and determine range R	Simulink libraries are used to reference the main model subsystems, leaving only the main body the weight for which is set to 50kg minus the weight of the subsystems plus battery.