Conceptual Modeling using Bond Graph as a Unified Meta-modeling framework.

Sampath Kumar Veera Ragavan,
School of Engineering, Monash University Sunway Campus,
Malaysia. veera.ragavan@monash.edu

Madhavan Shanmugavel
School of Engineering, Monash University Sunway Campus,
Malaysia. madhavan.shanmugavel@monash.edu

Velappa Ganapathy
School of Computer Science, SRM University, Kattankulathur
Tamil Nadu, India.

Bijan Shirinzadeh
Monash University, Clayton Campus, Australia.
bijan.shirinzadeh@monash.edu

***Abstract*—In this paper we discuss an alternative Model Based Design approach which can be used to bridge the concept-implementation gap commonly encountered during the design of complex, software- intensive multidomain systems also known as Cyber Physical Systems (CPS). CPS differs from other mechatronic systems due to the close coupling between the physical and computing systems. The focus is on getting the physics right- the rest is mathematics. Existing Model based Development techniques are predominantly software based and use UML for modeling complex engineering systems leading to difficulties in Model validation and verification and code generation. We propose a novel and unified approach based on BG-UMF; a bond graph based Meta Modeling framework as a practical and viable alternative to OMGs UML/SysML/OCL combination for meta modeling CPS. The power of the framework is highlighted through an example scenario: Conceptual Design and Development of a UAV.**

Keywords – Model driven development; Unified Modeling framework; Conceptual Model Development.

I. Introduction

This paper discusses the experiences of using a Model Based Design approach to bridge the concept-implementation gap prevalent in a new breed of complex, software- intensive mechatronic systems called as Cyber Physical Systems (CPS). CPS calls for a close coupling between the physical and computing domains: the focus is on getting the physics right- the rest is mathematics.

Model Based Design has of late become very popular [18]. In Model Driven Architecture (MDA) approach Models are developed through transformation across 3 stages of Platform dependency [17] : Platform Specific Model (PSM), Platform Independant Model (PIM) and Computationally Independant Model (CIM). While Model transformations from PIM to PSM and PSM to Platform specific code is possible in MDA using automatic code generation tools, CIM to PIM transformations are not possible. The main focus and contribution of this paper is advocating Bond graph for meta modeling of CIM to PIM transformation; to bridge the CIM - PSM Gap and exemplify Bond Graphs(BG) not only as a viable alternative but also formal method for domain independent modeling.

MDA mainly uses UML or its variants and extentions such as SysML, UML-RT. Object management group (OMG), expects UML will become a de-facto language for specifying, visualizing, constructing and documenting software system artifacts. In reality UML is found lacking in many aspects especially in semantics. Hence it has been augmented using other formal languages such as OCL for constraint propagation and maintain consistency between models There are many compatibility issues among UML variants making validation and verification extremely difficult.

The idea employed is simple yet appealing. By employing Bond Graph (BG) causality for constraint propagation and leveraging on a small footprint of 9 simple BG elements to represent physical system across multiple domains in detail, we compose these systems as *Unified Physical Systems* and automatically generate Models and DAEs.

We developed a novel and unified approach based on BG-UMF; a bond graph based Meta Modeling framework as a practical and viable alternative to OMGs UML/SysML/OCL combination for meta modeling CPS. The developed framework is collaborative and hierarchical encompassing all phases of design from Concept to Implementation leveraging on existing automatic code generation tools. In the conceptual design it provides the much needed support to explore the design space and analyze the effects of individual model decisions on the collaborative design.

Using the framework we demonstrate that BG-UMF bridges the gap between concept design and implementation code seamlessly. We demonstrate through this work, the strengths of BG-UMF for modeling multidisciplinary collaborative concept modeling for design space exploration. The power of the framework is highlighted through an example scenario: Model based design and development of a UAV.

II. Background and Rationale in using Bond Graph Grammars for Model Transformation

Model Based Development techniques has recently gained popularity as a result of invention of an abundance of modeling concepts, languages and Tools. They span multiple disciplines such as Industrial automation,Real-time systems, hardware-software Co design, Business Process engineering, Informational systems design such as Web2.0 etc. They are developed by many engineers working at various levels of abstraction concurrently in many systems and software development projects. Model transformations forms the heart and soul of Model driven development processes [17], [3], help re-usability [16]. Established standards for creating meta-models such as the Meta-Object Facility exist but, there is currently no mature foundation for specifying transformations among models [3] and there have not been any attempts to explicitly model transformation languages yet [7]. One of the major difficulties in reusing Model transformations is the structural differences between input meta-models [7].

A survey points to template-based, rule-based, triple graph grammars transformation paradigms , supported by developer implementations such as Atl, AToM, GReAT, Moflon, Qvt, Vmts, but in each implementation the transformation paradigm is hard-coded [7] and not reusable [16].They are based on Model Transformation in Discrete domain and Continuous Time or Hybrid models are difficult to transform using these tools.

Since 1970's the area of *Graph Transformation* a.k.a Graph Grammars or Graph Rewriting has grown in popularity as an independent branch mainly due to generalisation of string grammars and graph terms rewriting. Graphs have been used traditionally in Modelling as they are the simplest and universal models of representation not only in engineering and computer sciences but also in biological and life science models. Graphs are natural representations of models and many models are intrinsically graph based (e.g., statecharts, activity diagrams, collaboration diagrams, class diagrams, Petri nets). On the other hand Source Codes and Platform Specific Models are better represented using tree structures (e.g., parse trees, abstract syntax trees). Therefore we beleive Graph transformations are ideal to specify and execute model transformations in Computation and Platform independent (CIM and PIM) Levels of Model driven Design and Development [9].

As a science, Graph transformation synergistically merges the universal modeling paradigm of graphs with well-founded mathematically approaches to aid building models. In engineering, Graph Transformation provides the concepts, language rules and analytical tools for specification, modelling, and simulation for prediction, verification and validation of model properties and domain transformations.

In our BG-UMF application development framework, Bond Graph grammar not only provides the Formal Graph theoretical foundations for Model transformation it also provides a basis for Model based Design and Development. Advantages of this approach include:

1. Enabling abstract design of Functional Models at Higher levels
2. Providing a visual framework with powerful semantics and syntax to map Function and Behavior and between models [7].
3. Guaranteed Model Transformations from Concept to Code.
4. Providing a structural basis for transformation reuse.

The UML/OCL based methodologies for MDA primarily operate in discrete domain and hence are not very suitable for dynamic energetic systems. More over Model transformations and Formal Verification is not guaranteed. In this paper we propose BG-UMF an alternate methodology for MDA based on Continuous domain transformations.

III. A Bond Graph based Unified Meta-Modelling Framework (BG-UMF) for Design and validation of Dynamic Systems

Fig.1 shows the proposed General Purpose Modeling Framework which can be used for design and validation of Multi-domain, Nonlinear Engineering Systems. The framework is systematic and recursive and based on Bond Graph Modeling paradigm. It provides a *Unified, Hierarchical, Concurrent and Integrated* environment for design of *Dynamic Systems*. Executable models for *Virtual Prototype* and *Hardware in the Loop* Validation can be obtained from the framework. Moreover it can be used in all the three stages: *Concept*, *Detailed Design* and *Recursive Refinement* stages.

Using the Framework shown in Fig. 1 as a systematic design methodology we proceed to synthesise *Functional* and *Behavioral* aspects of System Design in stages:

Conceptual Design Stage

Very few tools are available to support conceptual design [21]. Characterization of a design prototype at this stage is based on *Form*, *Function* and *Behavior* [11] and as *Components* [15] using *Word Bond Graphs(WBG)* as shown in Fig. 2. Physical aspects are categorized under Form; Purpose of the module defines the Function ; and the desired output specifies the required Behavior. Existing Conceptual design tools cater more to *Form* and less to *Function* and *Behavior*. Our work focuses on Behavior and Functions to achieve Conceptual design. At the conceptual design stage, detailed models cannot be developed as many design specifications are still subject to changes and accurate parameter values are unknown. Simple relationships that describe the high-level behavior (*e.g.* topological maps), system level schematics and system specifications are available. A simulation at this stage depends more on the accuracy of the parameters available in the course grained topological model. Generation, modification and evaluation of the conceptual design can

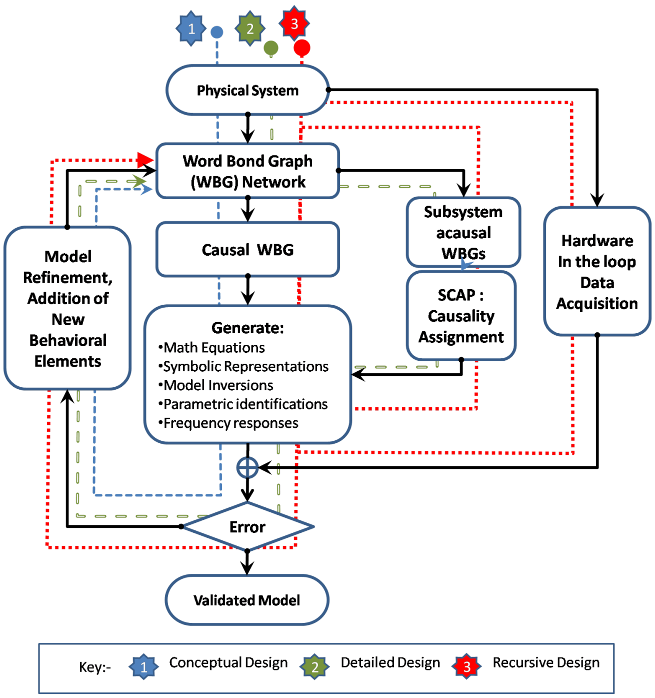

Fig. 1. **B**G-**UMF** - A **B**ond **G**raph based **U**nified **M**eta **M**odelling **F**ramework for Design and Validation of Dynamic Systems

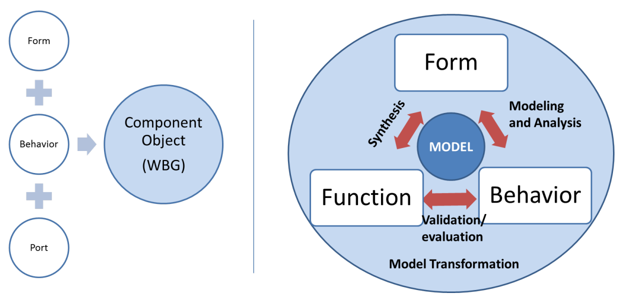

Fig. 2. Relation ship between *Form*, *Function*, *Behaviour* and *Component* in the Context of Word Bond Graphs (WBG). Form relates to physical aspects, Function to the purpose and Behaviour to the desired output.

be repeated until all design alternatives are exhausted. Main requirement for the models at this stage is that it should be easy to create, flexible to changes and remain relevant and concurrent throughout the design life cycle.

Word Bond Graph (WBG) can be used to provide a conceptual abstract overview. WBGs can be compiled even at conceptual level to obtain system equations help ascertain the estimated overall system performance. They are different from Block diagrams. While *Block Diagrams* also represent physical systems, they cannot be compiled at higher levels of abstraction. Block diagrams use assignment statements instead of equations and hence the Blocks cannot be compiled until the component input and output specifications are completely fixed [4]. Another advantage is that the WBGs can be used in a Bottom-up, Top-down or in iterative approaches.

A. *Detail Design Stage*

The most challenging part of the detailed design stage is to maintain the concurrency and integrity due to the presence of both *Discrete* and *Continuous* parameters [1]. At this level, Component Behaviour, Physical Phenomena, Structural Parameters and Form constraints become available for consideration.

B. *Recursive Refinement Stage*

Creating high fidelity simulation models is a complex recursive activity and highly time consuming. Requirements at this stage include reuse, integration with the design environment, flexibility, ease of use, intuitive interface and support for hierarchical decomposition [12]. The framework should also remain relevant even if the simulation tools become obsolete. The human approach to concept design has not changed much over the years. We starting from a "Single Concept" and improve the concept iteratively until the Requirements are met. We discard concepts only when they become unworkable and futile. Hence a major requirement for a General Unified framework is that each of these stages should be iterative and recursive, permitting evolution of Conceptual design.

IV. Discussion

Benefits of the proposed BG-UMF in the current context accrue from the inherent strengths of BG as a UML and the Hierarchical Design Methodology proposed by the Framework. Major benefits include:

1. Ability to predict system response even before physical prototype is assembled [5].
2. Computer simulation for predicting the system response and stability
3. Automated simulation for Nonlinear Systems. Advanced packages that can generate causality and state equations automatically are available today.
4. Concept Design Framework and a Unified Knowledge Representation Framework that can be carried throughout the product life cycle.
5. Estimate the overall achievable performance and propose an approach for the achievable performance problem.
6. Concurrent and Collaborative Design environment
7. Facilitate building Non Linear Models by extending linear models and Vice versa.
8. Easy linearisation of Nonlinear Kinematics allows developing an intuitive feeling which is hard to get from pure computer simulations.
9. Ability to test and validate control strategies developed using linear methods near boundaries.
10. Develop Standard Ontologies from BG that allows seamless and unambiguous communication between disparate design teams.

Benefits (1-5) listed above have been adopted and exemplified in this paper. Benefits (7-8) listed above have been exemplified in [13] and Benefits(9-10) will be covered in a future work.

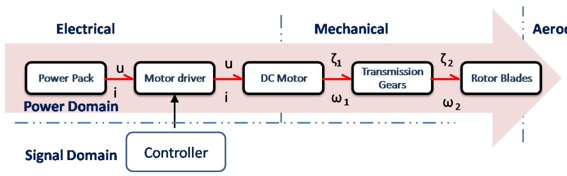

Fig. 3. Multi-domain Power Flow Visualisation: Power flows from the drive to the Propeller is shown. Power exchange takes place across three domains: Electrical, Mechanical and Aerodynamic domains. Power Exchange across domains are modelled easily in BG using Transformation elements (*Transformers (TR)* and *Gyrators (GY)*

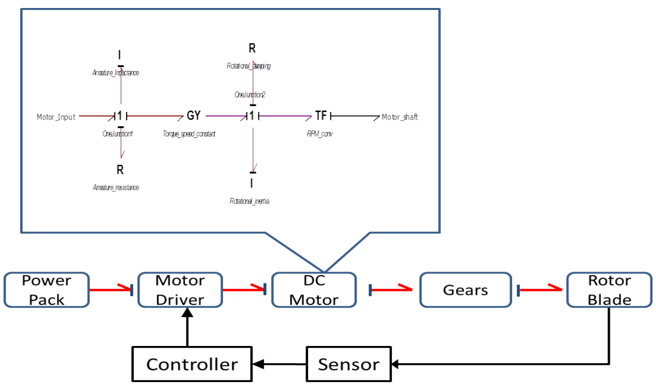

Fig. 4. Hierarchical Decomposition of Propeller Group Dynamics using Word Bond Graphs and System level Integrity check for constraint propagation using Causal assignments

V. Methodology: Steps for Concept Transformation into Composable Physical Models.

Using the Framework and the FBS Design methodology [20], we automate the design process from conceptual design to Code generation. The output of this Framework is a Platform Independent model which can be easily converted to platform specific models using tool chains such as Matlab/Simulink or NI Labview.

Once the *Concept* is available in the form of a *Domain Independent Topological Representation*, (e.g. as shown in Fig. 5(a)), the following methodology can be generally followed for transformation of *Concept Models* to *Ideal Physical Models*. These steps have been adapted from various approaches [2], [5] and [10]:

1. Functional Decomposition and Modularization
   - a Identify the Multiple energy domains to form Subsystems and Components.
   - b Couple subsystems of distinct but continuous Power domains using Linear/Non-Linear Power Conserving Transducers. (typically modelled as *TF / MTF* or *GY / MGY* elements in BG).
   - c Using Amplifiers and Instruments, couple discontinuous subsystems or when System descriptions are "Functional" rather than "Physical".
2. Embodiment of behavior and structure. FBS proposes Knowledge representation schemes to systematically decompose objective concept design functionality to subjective functions related to each other by a function-behavior relationship mainly from a mechanical systems design perspective. Even today there is no objective methodology or algorithm for functional decomposition [20]. Using the framework every function that has been decomposed objectively or subjectively has a unique behavioral relation, and can be mapped to a structure. Some functional entities that have more than one function or other functions can still be modeled by adding objectivity. (Example a fastner - Nut and Bolt cannot be modeled by BG as it has a structure that can imply many unintended applications. Once the functional intent is made explicit, then it can be modeled in a Bond Graph. If the purpose of a fastener is to apply a force to hold down another mass in place then its function is to apply a force (S_e). It is easier to extend the model to include dynamic behavior of the fastener such as vibration analysis etc) In other representations, the model becomes defunct when it is functionally overloaded where as in this framework, it can be extended to include a spring washer to the fastener and analyze the dynamics.
3. Achievable Performance Estimation : Introduce available Efficiency estimates using Linear/ Non-linear R elements for Performance analysis. For example wind resistance, wheel resistances etc can be modeled as a non-linear "R" elements proportional to the Velocity of rotation.
4. Unification - Behavioral and structural composition:
   - a Identify Junctions for building Junction Structure. Junctions have a special interpretation for each domain such as Connection in parallel / connection in series.
   - b Choose the Unified reference variables to map the power (*Effort* or *Flow*). Method of *Effort Mapping* or *Flow Mapping* is sufficient to model most domains.
   - a In *Effort Mapping* well defined Effort are represented by *0 - Junction* and Effort Differences by a *1- Junction.*
   - b In *Flow Mapping* well defined Flows represented by *1 - Junction* and Flow differences by a *0 Junction* .
5. Consistency check: Identify the port type and list the type of connection required i.e. effort

or flow (analogous to *Across* or *Through* type variables) Check Causality assignment to see if desired/preferred causality is maintained.

6. Automation: Perform Junction simplifications if necessary. Tools such as 20-SIM are available to remove superfluous junctions and generate State-Space models.

VI. Concept Design of a Micro UAV

In the context of this paper, Design is the Transformation from *Function* to *Form*. Conceptual design is the beginning of this transformation [19] where Form(structure) and Function though related do not uniquely map onto each other. Form(structure) can be specified in various levels of abstraction ranging from a napkin sketch to a scale model. In concept design transformation results in a form more typical of the initial sketch than a full scale model.

Design of Miniature Rotorcraft UAVs differ significantly from full scale Rotorcraft UAVs. Weight of propulsion group, Low Reynolds Number aerodynamics, Low velocities Flights, Varying speed drives and Fixed pitch rotors are some of the major influences differentiating the Concept design. Only recently due to reduced weight and higher Power of Propulsion group components and battery, other configurations such as co-axial, tandem, quad-rotors, hexa-rotors etc have come into existence [14]. Concept Design and Modeling of Micro UAV is ad-hock and most of the time data is unavailable for modeling. Currently prototype design is the only robust method to test and validate designs.

Two Concept Models developed iteratively using the framework follows. As the Propulsion Group is the most important functional group in UAV designs, we start with simple concept design as follows.

Initial Concept 1: uses a single rotor with simple aerodynamics, Thrust and Torque equations and 1-D bond graphs for a planar rigid body resulting in a simple 3 DOF model as shown in Fig 5 .

Extended Concept 2: uses a single rotor with option for additional rotors. Body forces and Moments and aerodynamics forces and combination of 1-D and 3-D bond Graphs to allow for 4 DOF Planar Model. Altitude control and heading decoupled. A simple Planar rigid body model is used. This model can be easily replaced with a 3D rigid body model for more detailed analysis. Continuity in design is not lost and detailed models with 3D Kinematics, and Dynamics, can be seamlessly developed and improved as explained in our earlier [13]

A. Developing a Simple Concept Model

The topological maps written using domain independent symbols (e.g. in Fig 5.(a)) can then be systematically transformed to idealphysical models, using the basic bondgraph elements and the bonds (power and signal).

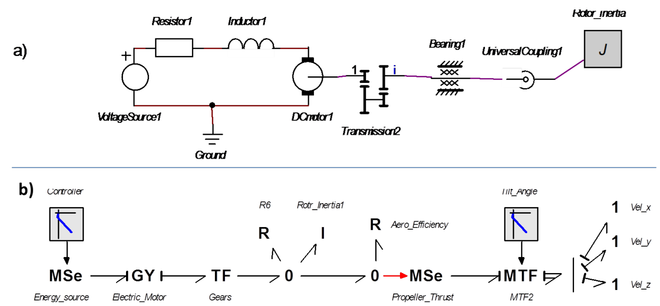

Fig. 5. Initial Concept Development - From Form to Behavior Transformation

1. Identification of the Power domains and elements: The Power domains in this case are Electrical, Mechanical rotation and Mechanical translation and Aerodynamics. Each Domain transformation is modeled usually using *TF or GY* elements. For example the Gear Box is modeled as a TF element, Electric Motor as a Gyrator. Aerodynamic Thrust and Moments are modeled as *MSe* element and *MTF*elements are used to Model Coordinate Transformation.
2. Identify the reference effort (or flow) for each of the domains listed with direction conventions. Force of gravity acts downward and thrust in the upward direction in the *Body Frame*.
3. Identification of junctions and building a Junction Structure:

B. Advanced Concept Model

To improve upon the initial concept model we sequentially add functions to increase the quality and complexity of the model without loss of continuity as detailed below. The advanced Concept Model is shown in Fig 7.

1) Planar Rigid Body Dynamics: In order to model UAV flying at constant altitude, a simple planar model of a rigid body is normally used.

Fundamentally, Bond graphs rely on linear and angular Momentum variables rather than linear and angular acceleration [6]. Fig 6 shows the 4 DOF's of a Planar Rigid Body (v_x, v_y, v_z, ψ), in planar Motion. For the sake of analysis the Propeller is assumed to Rotate at constant angular velocity and produce a constant thrust in the Z direction. The difference in the Angular velocities of the propellers produce a Moment τ thereby steering the UAV in the $X - Z$ plane. Tilting the Rotor causes Thrust in F_x and F_y directions resulting in a velocity in the respective directions. This Model of Rigid Body is very similar in concept to the General Plane Motion in Body Fixed Coordinates as discussed in [5, Pg.110]

The *Body Fixed* Velocity components of v_x, v_y, ψ are used instead of the Inertial velocities V_X, V_Y, and Ψ where

ψ is the rotation about z-axis given by the following Newton-Euler Equations:

$$\sum F_x + m\psi v_y = m\dot{v_x} \tag{1}$$

$$\sum F_y - m\psi v_x = m\dot{v_y} \tag{2}$$

The above Planar Rigid Body equations can be represented in Bond graphs using three *1- Junctions* one for each of the three degrees of freedom x, y, θ connected by a *MGY* element to couple the force component in the x-direction ($m\omega$ times) with the velocity in the y-direction and vice versa. The Planar Rigid Body Motion can be represented in BG as shown in Fig. 6. The Thrust in the z-direction is assumed to be uncoupled (constant altitude) and the angular rate ψ is assumed to be constant during simulation.

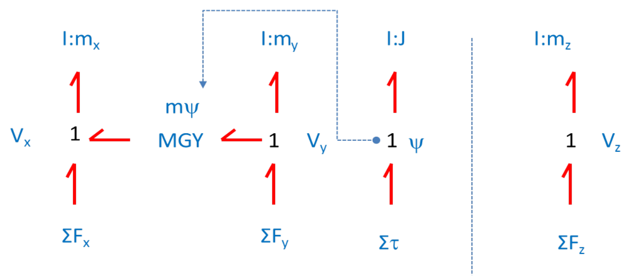

Fig. 6. Planar Motion in Body Fixed Coordinates. 4 DOFs represented by *1-Junctions* and the Coupling through *GY* element.

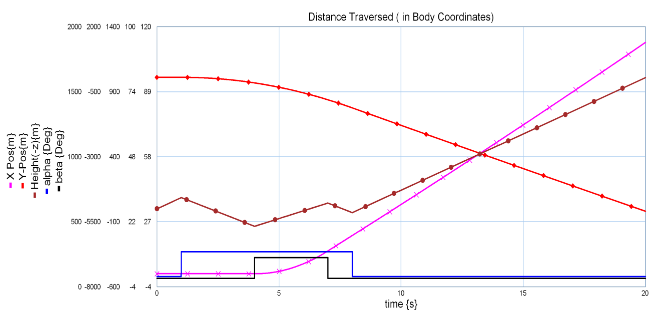

Fig. 9. Displacement In Body Coordinate Frame of UAV

2) Aerodynamics: The External forces and moments acting on the UAV as given by [14], [8] are :

$$F = T_{up} + W_{fus} + F_g \tag{3}$$

where Thrust magnitude T_i of a Rotor i of radius R is given by

$$T_i = C_{Ti}\pi\rho R^4\Omega_i^2 = C_{Ti}k_T\Omega_i^2 \tag{4}$$

where $k_T = \pi\rho R^4$. Aerodynamic drag is given by $W_{fus,i} = \frac{1}{2}C_{D,i}\rho A_i v_i^2$. The effect of Rotor rpm on the thrust as given by the above formula is plotted in Fig. 10.

Moments acting on the UAV is given by

$$M = Q_{up} + Q_{react,up} + M_{flap,up} + r_{Cup} \times T_{up} \tag{5}$$

where Torque magnitude Q_i of a Rotor of radius R is $Q_i = C_{Qi}\pi\rho R^5\Omega_i^2 = C_{Qi}k_Q\Omega_i^2$ where $k_Q = \pi\rho R^5$. These equations can be seen in the Aerodynamic Block in Fig. 7.

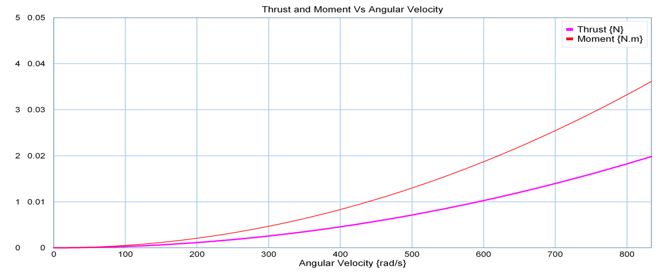

Fig. 10. Aerodynamic Thrust and Torque developed by Propeller

3) Direction Control: Apart from the magnitude of Thrust and Rotor torques, the Cyclic Pitch input is used to steer the helicopter by tilting the Rotors Tip Path plane (TPP). Direction of the thrust generated is perpendicular to TPP and defined using the tilt angles α and β about the x and y axis respectively.

$$T_i = T_i . n_{T,i} \tag{6}$$

$$n_{T,i} = \begin{bmatrix} \cos\alpha\sin\beta \\ \sin\alpha \\ -\cos\alpha\cos\beta \end{bmatrix} \tag{7}$$

The TPP tilt angles α and β produce Velocity Components in many directions and result in change of velocity and position in X, Y, and Z directions. Plot of velocity in Body Coordinate Frame of UAV in Fig. 8 shows the change in Velocity due to change in TPP angles (α, β). Change in TPP results in Thrust in the (x, y, z) as observable from the change in (x, y, z) velocities. The representation of TPP can be seen in the Transformation Block in Fig. 7

4) Gravity Effects: Modeling Gravity effects involves Coordinate transformations. The force of gravity constantly

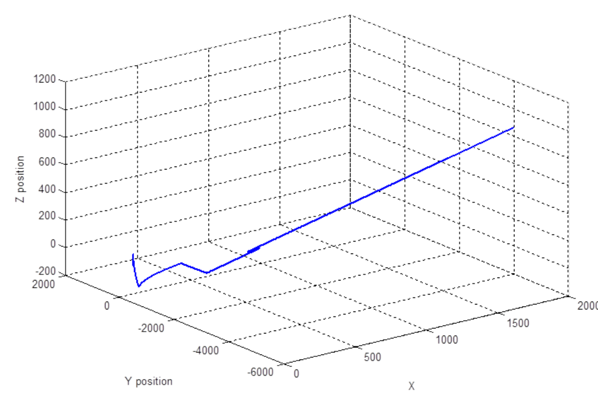

Fig. 11. Plot of UAV Path as a result of changes in TPP angles (α, β). UAV traveling from an initial coordinate $(100, 50, 0)$ to final Position of $(1800, 5000, 1000)$ as a result of change in TPP angles

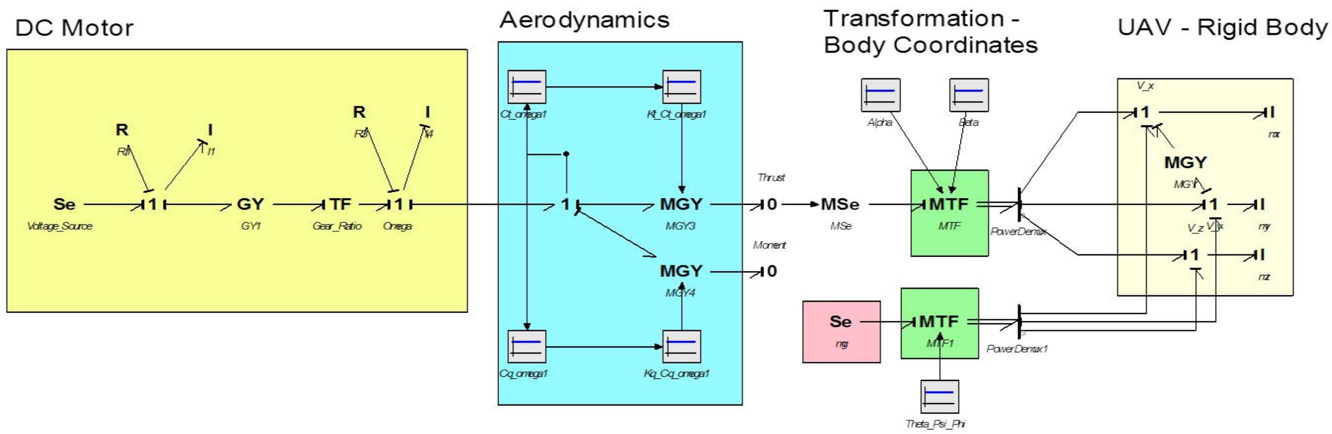

Fig. 7. Detailed Concept showing Multi Domain power Flow: Power flows from the Drive to the Rotor (propeller). In the process energy exchange takes place across 3 Domains- electric, Mechanical and Aerodynamic domains.

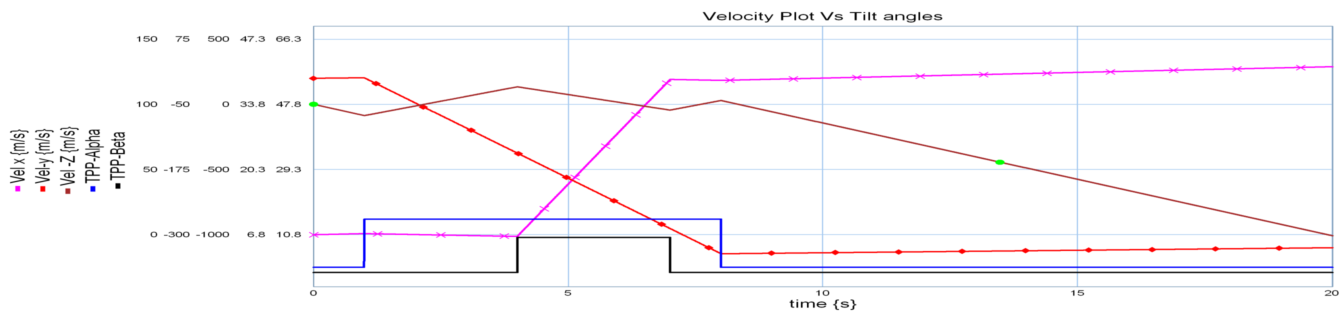

Fig. 8. Plot of velocity in Body Coordinate Frame of UAV: A change in TPP angles (α, β) resulting in Thrust in the (x, y, z) as seen from the change in (x, y, z) velocities

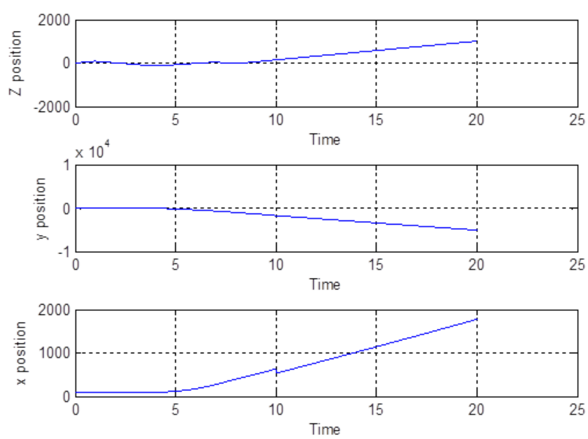

Fig. 12. Velocity Plot

acts in Z direction in the Inertial frame where as the Body Coordinate Frame (BCF) and orientation of the aircraft is different. A Transformation from Inertial frame to the Body

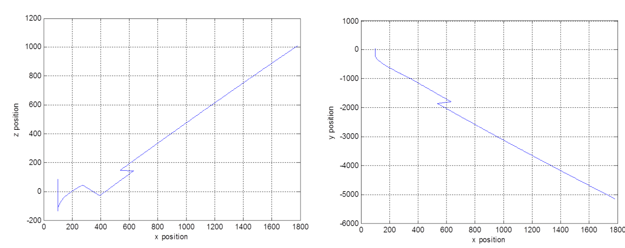

Fig. 13. Path of UAV in XY and XZ planes

Coordinate Form and Vice versa is done using MTF elements. Force of gravity due to the helicopter mass $F_G = mg$ acts constantly in Z direction in the inertial frame, which has been transformed to the BCF using the *MTF* element:

$$G = mg \begin{bmatrix} -\sin\theta \\ \sin\phi\cos\theta \\ \cos\phi\cos\theta \end{bmatrix} \quad (8)$$

The Gravity force component can be seen in the bottom green MTF Block in Fig. 7.

5) Simulation and Testing: The Detail Model based on the above modeling considerations is shown in Fig. 7. A Compilation of the Model using 20-SIM^{TM} resulted in 66 equations, 60 Variables and 8 independent states. Various plots obtained show that the behavior of the UAV is on expected lines and the model is Qualitatively correct. Lastly, Fig. 11 shows a 3D plot the UAV traveling from an initial coordinates $(100, 50, 0)$ as a result of Tilt angles to reach a final Position of $(1800, 5000, 1000)$.

VII. Conclusion

In this paper a Bond Graph based Unified Modelling Framework (BG-UMF) for a general class of Mechatronic systems was proposed. The hypothesis of this paper that BG-UMF offers not only an alternative design framework to UML, but also provides the designer or analyst a comprehensive framework with excellent tools for understanding the dynamics of multi domain systems. It provides an ideal design and development platform for Conceptual Modelling of complex hybrid dynamic systems such as UAVs. They provide designers, modellers and inventors immense physical insight that is necessary to prevent mistakes and cost overruns.

By allowing a component framework and classification of subsystems using FBS design methodology, a systematic top-down and bottom-up model formulation has been enabled. This increases knowledge sharing and improves the life cycle times. We demonstrate the agility and reuse of the modelling approach by breaking the system down to subsystems before performing systemic or phenomenological analysis. In a top-down approach, the models can be used to dimension the components of UAV and to optimise the physical and inertial parameters.

In case of a bottoms-up approach the qualitative experimental models can be used to correlate the physical and behavioural aspects of the UAV and to recursively refine the system. The Meta modelling support available is expected to keep the framework relevant during the systems life-cycle. The Framework serves as an unifying vehicle to analyse and transform knowledge and information coherently throughout the systems life cycle.

As discussed, using the proposed framework, a high fidelity, Non Linear dynamic model that accurately describes the UAV in flight can been developed and validation performed through simulation details of which are available in our earlier paper [13]. System response generated using 20-SIM provides adequate insight for Performance assessments and Control system design.

References

[1] S. Behbahani and C. de Silva, "Mechatronic design quotient as the basis of a new multicriteria mechatronic design methodology," *Mechatronics, IEEE/ASME Transactions on*, vol. 12, no. 2, pp. 227–232, 2007.

[2] P. Breedveld, "Concept-oriented modeling of dynamic behavior," *Bond Graph Modelling of Engineering Systems: Theory, Applications and Software Support*, p. 1, 2011.

[3] K. Czarnecki and S. Helsen, "Feature-based survey of model transformation approaches," *IBM Systems Journal*, vol. 45, no. 3, pp. 621–645, 2006.

[4] P. Gawthrop and G. Bevan, "Bond-graph modeling," *Control Systems Magazine, IEEE*, vol. 27, no. 2, pp. 24–45, 2007.

[5] D. Karnopp, D. Margolis, and R. Rosenberg, *System dynamics: modeling and simulation of mechatronic systems*. John Wiley & Sons New Jersey, 2006, vol. 3.

[6] D. Karnopp, *Vehicle Dynamics, Stability, and Control*. CRC Press, 2013, vol. 221.

[7] T. Kühne, G. Mezei, E. Syriani, H. Vangheluwe, and M. Wimmer, "Explicit transformation modeling," in *Models in Software Engineering*. Springer, 2010, pp. 240–255.

[8] J. Leishman, *Principles of helicopter aerodynamics*. Cambridge Univ Pr, 2006.

[9] T. Mens and P. Van Gorp, "A taxonomy of model transformation," *Electronic Notes in Theoretical Computer Science*, vol. 152, pp. 125–142, 2006.

[10] A. Mukherjee, *Bond graph in modeling, simulation and fault identification*. CRC Press, 2006.

[11] G. Pahl, W. Beitz, and K. Wallace, *Engineering design: a systematic approach*. Springer Verlag, 1996.

[12] C. Paredis, A. Diaz-Calderon, R. Sinha, and P. Khosla, "Composable models for simulation-based design," *Engineering with Computers*, vol. 17, no. 2, pp. 112–128, 2001.

[13] S. V. Ragavan, M. Shanmugavel, B. Shirinzadeh, and V. Ganapathy, "Unified modelling framework for uavs using bond graphs," in *Intelligent Systems Design and Applications (ISDA), 2012 12th International Conference on*. IEEE, 2012, pp. 21–27. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6416507

[14] D. Schafroth, C. Bermes, S. Bouabdallah, and R. Siegwart, "Modeling and system identification of the mufly micro helicopter," *Journal of Intelligent & Robotic Systems*, vol. 57, no. 1, pp. 27–47, 2010.

[15] C. Schmitke and J. McPhee, "Forming equivalent subsystem components to facilitate the modelling of mechatronic multibody systems," *Multibody System Dynamics*, vol. 14, no. 1, pp. 81–110, 2005.

[16] S. Sen, N. Moha, V. Mahé, O. Barais, B. Baudry, and J.-M. Jézéquel, "Reusable model transformations," *Software & Systems Modeling*, vol. 11, no. 1, pp. 111–125, 2012.

[17] S. Sendall and W. Kozaczynski, "Model transformation: The heart and soul of model-driven software development," *Software, IEEE*, vol. 20, no. 5, pp. 42–45, 2003.

[18] D. Thomas, "Mda: Revenge of the modelers or uml utopia?" *Software, IEEE*, vol. 21, no. 3, pp. 15–17, 2004.

[19] K. Ulrich and W. Seering, "Computation and conceptual design," *Robotics and Computer-Integrated Manufacturing*, vol. 4, no. 34, pp. 309 – 315, 1988, ¡ce:title¿Special Issue Manufacturing Systems and Technology of the Future¡/ce:title¿. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0736584588900026

[20] Y. Umeda, H. Takeda, T. Tomiyama, and H. Yoshikawa, "Function, behaviour, and structure," *Applications of artificial intelligence in engineering V*, vol. 1, pp. 177–194, 1990.

[21] N. Vargas-Hernandez, J. Shah, and Z. Lacroix, "Development of a computer aided conceptual design tool for complex electromechanical systems," in *Computational Synthesis: From Basic Building Blocks to High Level Functionality, Papers from the 2003 AAAI Symposium Technical Report SS-03-02*, 2003, pp. 255–261.