2002

Smart flight control surfaces with microelectromechanical systems

Sergey Lyshevski

Follow this and additional works at: http://scholarworks.rit.edu/article

Recommended Citation

http://scholarworks.rit.edu/article/286

This Article is brought to you for free and open access by RIT Scholar Works. It has been accepted for inclusion in Articles by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.
Smart Flight Control Surfaces  
With Microelectromechanical Systems

SERGEY EDWARD LYSHEVSKI  
Purdue University

Pitch, roll, and yaw moments can be developed by deflecting and changing the geometry of control surfaces. In this paper, smart flight control surfaces are designed using multi-node microelectromechanical systems (MEMS) to displace control surfaces and change the surface geometry. These MEMS augment translational motion microstructures (actuators-sensors), controlling/signal processing integrated circuits (ICs), radiating energy devices and antennas. The desired pitch, roll, and yaw moments are produced, drag can be reduced, and unsteady aerodynamic flows are controlled by smart flight control surfaces. That is, we achieve aerodynamic moment and active flow control capabilities. The major objective here is to report fundamental and applied research in design of smart flight control surfaces with MEMS-based actuator-sensor-IC arrays controlled by hierarchical distributed systems. We demonstrate the feasibility and effectiveness of the application of smart flight control surfaces for coordinated longitudinal and lateral vehicle control.

I. INTRODUCTION

The application of smart structures and microelectromechanical systems (MEMS) in design of flight control surfaces is reported in [1—4]. Conventional hydraulic and electromechanical actuators are widely used to actuate aircraft control surfaces if high torque is needed [5—7]. In micro air vehicles, smaller deflections and torques are required, and MEMS technology is uniquely suitable to attain active aerodynamic control. In this paper, we design smart flight surfaces using the array of MEMS nodes controlled by the distributed controller. Point-to-point actuation or sensing, performed by microstructures is not sufficient, and real-time intelligent coordinated motion with complex surface geometry synthesis is frequently required. Smart control surfaces can be built using thousands of MEMS nodes. These motion microdevices must be interfaced and controlled using integrated circuits (ICs) [8, 9]. Decentralized control systems with increased level of integration and functionality must be designed. The application of multiscalar digital signal processors (DSPs) allows one to design distributed control systems to control MEMS arrays. Distributed embedded control systems offer several advantages because affordable systems can be designed using low-cost high-performance MEMS nodes. Energy and control signals are transmitted and distributed within the microdevises (actuators-sensors) through power, communication and control channels. The distributed systems must have distributed processor capacity. The communication between different nodes in distributed control systems consists of data and administrative messages. Therefore, the complexity of distributed systems is higher than the complexity of centralized control systems. The communication between the supervising high-level layer and lower level layers is based upon access messages, set-up messages, requests and responses, as well as status, events, and error messages. Hence, in distributed systems, the processor capacity is distributed within the system. Each node performs the specific functions (tasks), and distributed systems are organized in different ways as the designer synthesizes the overall system and defines the node functions and tasks to guarantee the desired functionality and operationability. In this article, we synthesize smart flight surfaces using MEMS and design the distributed control system.

Control of smart flight surfaces is performed by the hierarchical distributed management system to properly displace microstructures, and thus to change the geometry of control surfaces in order to guarantee the desired moments, control the aerodynamic flows, and reduce the drag. The design method and system architecture are different compared with conventional systems because the benefits and capabilities of smart flight surfaces can be utilized through hierarchical
decentralized control and adaptation with performance analysis and outcomes prediction. It is illustrated that the integration and interfacing of high-, medium- and low-level layers are critical to attaining desired performance.

The mathematical model of smart flight surfaces is developed using the distributed actuators-sensors array architecture, and MEMS nodes are modeled applying electromechanics and tensor analysis. These MEMS nodes (motion microdevices—ICs) operate under uncertain rapidly evolving environment. Nonlinear dominant dynamics of MEMS is modeled. Flight vehicles operate at wide range of flight conditions (different velocity, angle-of-attack, pressure, temperature, etc.), scenarios, and missions. Fast and unmodeled dynamics, parameter variations, uncertainties and other secondary phenomena are integrated in the analysis, modeling and design by applying robust control theory.

We study the application of MEMS for active aerodynamic control including unsteady aerodynamics. Therefore, very fast dynamics of motion microdevices must be guaranteed. The MEMS nodes are controlled using the closed-loop system. Control surfaces are actuated displacing high-performance microactuators. Highly robust and reliable MEMS-based shear stress sensors are capable of accurately sensing the steady and unsteady aerodynamic loads through the flight surface allowing us to guarantee optimal aerodynamic control based upon analysis and control of aerodynamic flow and loads. The reported concept allows the designer to decrease the size and number of control surfaces needed, relaxing the spectrum of complex and conflicting requirements. For micro air vehicles, using the MEMS-based smart flaps, perfectly coordinated longitudinal and lateral control can be achieved in the full operating envelope, and the needed rolling, pitching and yawing moments are developed. A MEMS-based prototype of smart flaps is designed and tested through high-fidelity nonlinear modeling, data-intensive analysis and heterogeneous simulation in order to demonstrate the efficiency of the reported concept. Optimal flight vehicle aerodynamic control improves vehicle performance (flying characteristics, controllability, agility, maneuverability, etc.) which cannot be achieved using conventional flight actuators.

II. FLIGHT VEHICLES AND MEMS ARRAY

The local aerodynamic flow is actively controlled and sensed by the specific microstructure (MEMS node), and MEMS-based actuator-sensor-ICs arrays are designed in order to deflect and change the geometry of smart flight surfaces. Different MEMS configurations and architectures can be designed for different flight control surfaces (ailerons, elevators, fins, flaps, stabilizers and tips) analyzing the forces and moments need to be developed, deflection angle, deflection rate, surface geometry, aerodynamic loads, drag, as well as other specifications. However, for all control surfaces, hierarchical distributed closed-loop systems are needed to be designed in order to guarantee the coordinated longitudinal and lateral vehicle control as well as to ensure the active aerodynamic flow control. It is shown that despite of complexity of flight management system architectures and control algorithms, hierarchical decentralized systems can be designed to guarantee the specifications and requirements imposed on flight vehicles. Ailerons, elevators, canards, fins, flaps, rudders, stabilizers and tips can be controlled by MEMS. For micro air vehicle we consider the smart flight surfaces (two flaps) which are designed using MEMS-based actuator-sensor-ICs arrays in order to perform coordinated vehicle control by displacing surfaces, changing the surfaces geometry, and sensing the aerodynamic loads, see Fig. 1.

The MEMS array is built using MEMS nodes, and a single node is illustrated in Fig. 2 [9].

The primary focus of this article is the design of smart flight surfaces with MEMS-base actuator-sensor-ICs arrays controlled by the hierarchical distributed closed-loop system. Three-layer hierarchically distributed system architecture for is documented in Fig. 3.
III. NONLINEAR DYNAMICS OF FLIGHT VEHICLES

For flight vehicles, the application of the Lagrangian mechanics results in the following state-space model [10–12]

\[
\begin{bmatrix}
\dot{v}(t) \\
\dot{\alpha}(t) \\
\dot{q}(t) \\
\dot{\theta}(t) \\
\dot{\beta}(t) \\
\dot{p}(t) \\
\dot{r}(t) \\
\dot{\phi}(t) \\
\dot{\psi}(t)
\end{bmatrix} = A \begin{bmatrix}
0 \\
-p \cos \alpha \tan \beta - r \sin \alpha \tan \beta \\
\frac{1}{I_y} (I_Z - I_Y) \rho r - I_X Z p^2 + I_X Z r^2 \\
q \cos \phi - r \sin \phi \\
p \sin \alpha - r \cos \alpha \\
\frac{1}{I_X Z - I_{ZX}} (I_X Z (I_X - I_Y + I_Z) q p + (I_Y Z - I_{ZX} - I_{XX} Z) q r) \\
\frac{1}{I_X Z - I_{ZX}} (I_X Z (I_X - I_Y + I_Z) q p - I_X Z (I_X - I_Y + I_Z) q r) \\
q \tan \theta \sin \phi + r \tan \theta \cos \phi \\
q \cos^{-1} \theta \sin \phi + r \cos^{-1} \theta \cos \phi
\end{bmatrix} + \begin{bmatrix}
F_v \\
F_\alpha \\
M_M \\
I_Y \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
\]

where \( v \) is the forward velocity; \( \alpha \) is the angle-of-attack; \( \beta \) is the sideslip angle; \( q, p \), and \( r \) are the pitch, roll, and yaw rate; \( \theta, \phi, \) and \( \psi \) are the elevation, bank, and azimuth angles (the Euler angles); \( m_\alpha \) is the mass; \( F_v, F_\alpha, \) and \( F_\beta \) are the applied engine, aerodynamic and gravitational forces acting on the airframe; \( M_M, M_L, \) and \( M_N \) are the pitching, rolling and yawing aerodynamic moments which are functions of the control surface deflection and geometry; \( I_X, I_Y, I_Z, \) and \( I_{XX} X Z \) are the moments of inertia; \( A \in \mathbb{R}^{9 \times 9} \) is the matrix of coefficients.

IV. ACTUATOR-SENSOR DYNAMICS

A complete electromagnetic model of MEMS is derived in terms of five electromagnetic field vectors, three electric field vectors, and two magnetic field vectors. The electric field vectors are the electric field intensity \( \vec{E} \), the electric flux density \( \vec{D} \), and the current density \( \vec{J} \). The magnetic field vectors are the magnetic field intensity \( \vec{H} \) and the magnetic field density \( \vec{B} \).

The differential equations for motion microdevices are found using Maxwell’s equations [9], constitutive (auxiliary) equations, and classical mechanics. The Maxwell’s partial differential equations in the \( \vec{E} \) - and \( \vec{H} \) -domain in the point form are

\[
\nabla \times \vec{E}(x,y,z,t) = -\mu \frac{\partial \vec{H}(x,y,z,t)}{\partial t}
\]

\[
\nabla \times \vec{H}(x,y,z,t) = \varepsilon \frac{\partial \vec{E}(x,y,z,t)}{\partial t} + \vec{J}(x,y,z,t)
\]

\[
\nabla \cdot \vec{E}(x,y,z,t) = \frac{\rho_e(x,y,z,t)}{\varepsilon}
\]

\[
\nabla \cdot \vec{H}(x,y,z,t) = 0
\]

where \( \varepsilon \) is the permittivity; \( \mu \) is the permeability; \( \sigma \) is the conductivity; \( \rho_e \) is the volume charge density.

The constitutive (auxiliary) equations are found using the permittivity \( \varepsilon \), permeability tensor \( \mu \), and conductivity \( \sigma \). In particular,

\[
\vec{D} = \varepsilon \vec{E} \quad \text{or} \quad \vec{D} = \varepsilon \vec{E} + \vec{P},
\]

\[
\vec{B} = \mu \vec{H} \quad \text{or} \quad \vec{B} = \mu (\vec{H} + \vec{M}) \quad \text{and}
\]

\[
\vec{J} = \sigma \vec{E}.
\]

The Maxwell’s equations can be solved using the boundary conditions. We have

\[
\vec{a}_N \times (\vec{E}_2 - \vec{E}_1) = 0, \quad \vec{a}_N \times (\vec{H}_2 - \vec{H}_1) = \vec{J}_s,
\]

\[
\vec{a}_N \cdot (\vec{D}_2 - \vec{D}_1) = \rho_s, \quad \vec{a}_N \cdot (\vec{B}_2 - \vec{B}_1) = 0
\]

where \( \vec{J}_s \) is the surface current density vector; \( \vec{a}_N \) is the surface normal unit vector at the boundary from region 2 into region 1; \( \rho_s \) is the surface charge density.
The constitutive relations that describe media can be integrated with the Maxwell’s equations which relate the fields in order to find two partial differential equations. Using the electric and magnetic field intensities $\vec{E}$ and $\vec{H}$ to model electromagnetic fields in MEMS, one has

$$\nabla \times (\nabla \times \vec{E}) = \nabla(\nabla \cdot \vec{E}) - \nabla^2 \vec{E} = -\mu \frac{\partial \vec{J}}{\partial t} - \mu \varepsilon \frac{\partial^2 \vec{D}}{\partial t^2}$$

$$\nabla \times (\nabla \times \vec{H}) = \nabla(\nabla \cdot \vec{H}) - \nabla^2 \vec{H} = -\mu \sigma \frac{\partial \vec{H}}{\partial t} - \mu \varepsilon \frac{\partial^2 \vec{A}}{\partial t^2}.$$

The following pair of homogeneous and inhomogeneous wave equations

$$\nabla^2 \vec{E} - \mu \sigma \frac{\partial \vec{E}}{\partial t} - \mu \varepsilon \frac{\partial^2 \vec{E}}{\partial t^2} = \nabla \left( \frac{\rho_0}{\varepsilon} \right)$$

$$\nabla^2 \vec{H} - \mu \sigma \frac{\partial \vec{H}}{\partial t} - \mu \varepsilon \frac{\partial^2 \vec{H}}{\partial t^2} = 0$$

is equivalent to four Maxwell’s equations and constitutive relations. For some cases, these two equations can be solved independently. It must be emphasized that it is not always possible to use the boundary conditions using only $\vec{E}$ and $\vec{H}$, and thus, the problem cannot always be simplified to two electromagnetic field vectors. Therefore, the electric scalar and magnetic vector potentials are used. Denoting the magnetic vector potential as $\vec{A}$ and the electric scalar potential as $V$, we have

$$\nabla \times \vec{A} = \vec{B} = \mu \vec{H} \quad \text{and} \quad \vec{E} = -\frac{\partial \vec{A}}{\partial t} - \nabla V.$$

The electromagnetic field is the derivative of the vector potential. Using the Lorentz equation

$$\nabla \cdot \vec{A} = -\frac{\partial V}{\partial t}$$

the inhomogeneous vector potential wave equation to be solved is

$$-\nabla^2 \vec{A} + \mu \sigma \frac{\partial \vec{A}}{\partial t} + \mu \varepsilon \frac{\partial^2 \vec{A}}{\partial t^2} = -\mu \sigma \nabla V.$$

To model motion microstructures, the mechanical equations must be used, and Newton’s second law is usually applied to derive the equations of motion.

Using the volume charge density $\rho_v$, the Lorenz force, which relates the electromagnetic and mechanical phenomena, is found as

$$\vec{F} = \rho_v(\vec{E} + \vec{v} \times \vec{B}) = \rho_v \vec{E} + \vec{J} \times \vec{B}.$$

The electromagnetic force is found applying the Maxwell stress tensor. This concept employs a volume integral to obtain the stored energy, and the stress at all points of a bounding surface can be determined.

The sum of the local stresses gives the net force. In particular, the electromagnetic stress is

$$\vec{F} = \int (\rho_v \vec{E} + \vec{J} \times \vec{B}) \, dv = \frac{1}{\mu} \int \vec{T} \cdot d\vec{s}.$$

The electromagnetic stress energy tensor (the second Maxwell stress tensor) can be found. In general, the electromagnetic torque developed can be found using the electromagnetic field. In particular, the electromagnetic stress tensor is given as

$$\vec{T} = \vec{T}_E + \vec{T}_M = \begin{bmatrix} E_1 D_1 & E_1 D_2 & E_1 D_3 \\ E_2 D_1 & E_2 D_2 & E_2 D_3 \\ E_3 D_1 & E_3 D_2 & E_3 D_3 \end{bmatrix} + \begin{bmatrix} B_1 H_1 - \frac{1}{2} B_1 H_2 & B_1 H_2 & B_1 H_3 \\ B_2 H_1 & B_2 H_2 - \frac{1}{2} B_2 H_3 & B_2 H_3 \\ B_3 H_1 & B_3 H_2 & B_3 H_3 - \frac{1}{2} B_3 H_3 \end{bmatrix}.$$

Having derived the force, Newton’s second law is straightforwardly applied to find the torsional-mechanical dynamics.

V. SMART CONTROL SURFACE DYNAMICS

To derive the mathematical model for the motion microsurface, we study the surface deflection $d(t,x,y)$ in the $xy$ plane. The mass of the undeflected surface per unit area $\rho$ is constant, and the membrane is perfectly flexible. For small deflections, the tension $T$ is the same at all points in all directions. The forces acting at the midpoints of the sides are $F_x = T \Delta x$ and $F_y = T \Delta y$. The forces $F_x$ and $F_y$ are tangential to the surface. The horizontal components of the forces are the cosine functions of the inclination angles. The horizontal components at the opposite sides (right and left) are equal because angles $\alpha$ and $\beta$ are small. The rectangular control microsurface is shown in Fig. 4 [9].

Making use of Newton’s second law, the partial differential equation is found to be

$$\frac{\partial^2 d(t,x,y)}{\partial t^2} = \frac{T}{\rho} \left( \frac{\partial^2 d(t,x,y)}{\partial x^2} + \frac{\partial^2 d(t,x,y)}{\partial y^2} \right) - F(t,x,y) \text{ or}$$

$$\frac{\partial^2 d(t,x,y)}{\partial t^2} = \frac{T}{\rho} \nabla^2 d(t,x,y) - F(t,x,y).$$
Using the initial and boundary conditions, the solution is found. Let the initial conditions be
\[
d(t_0, x, y) = d_0(x, y) \quad \text{and} \quad \frac{\partial d(t_0, x, y)}{\partial t} = d_1(x, y).
\]
Thus, the initial displacement \(d_0(x, y)\) and initial velocity \(d_1(x, y)\) are given.

Assume that the boundary conditions are \(d(t, x_0, y_0) = 0\) and \(d(t, x_f, y_f) = 0\).

Then, the solution is
\[
d(t, x, y) = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} d_{ij}(t, x, y) = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \left( A_{ij} \cos \lambda_{ij} t + B_{ij} \sin \lambda_{ij} t \right) \times \sin \frac{i \pi x}{a} \sin \frac{j \pi y}{b}\]
where the eigenvalues (characteristic values) are
\[
\lambda_{ij} = \sqrt{\frac{1}{a^2} \left( \frac{i \pi}{a} \right)^2 + \frac{1}{b^2} \left( \frac{j \pi}{b} \right)^2}.
\]

Using the initial conditions, the Fourier coefficients are found to be
\[
A_{ij} = \frac{4}{ab} \int_0^b \int_0^a d_0(x, y) \sin \frac{i \pi x}{a} \sin \frac{j \pi y}{b} \, dx \, dy \quad \text{and} \quad B_{ij} = \frac{4}{ab \lambda_{ij}} \int_0^b \int_0^a d_1(x, y) \sin \frac{i \pi x}{a} \sin \frac{j \pi y}{b} \, dx \, dy.
\]
That is, the Fourier coefficients are analytically derived in the form of the double Fourier series.

The hierarchical distributed systems must be designed for flight vehicles with smart flight surfaces in order to perform a number of complex functions and tasks in an uncertain evolving dynamic environment. In particular, our goal is the synthesis of control algorithms and architectures which maximize performance and efficiency minimizing system complexity through the following:

1) intelligence, learning, evolution, and adaptation with performance analysis and outcome prediction,
2) coordination and autonomy through tasks and functions generation, organization, and decomposition,
3) efficiency and stability in the full operating and mission envelopes,
4) real-time adaptation, reconfiguration, and diagnostics,
5) fault tolerance and robustness.

Control theory and engineering practice in the design and implementation of hierarchical hybrid (digital and analog subsystems are integrated, discrete, and continuous events are augmented) real-time large-scale systems have not been matured. Synthesis of optimal controllers for single MEMS are performed using conventional methods, e.g., the Hamilton-Jacobi theory, Lyapunov’s concept, maximum principle, and dynamic programming [11]. However, these methods do not allow the designer to attain the desired features for large-scale systems even though some methods (e.g., adaptive control, fuzzy logic, and neural networks) ensure optimization, adaptation, and reconfiguration. Thus, hierarchical architectures are needed to be designed and optimized to achieve intelligence, learning, evolution, adaptation with performance analysis and outcome prediction in rapidly changing environments.

The design of intelligent adaptive closed-loop systems is formulated as a search problem in high-dimensional space, and the performance criteria form hypersurfaces. Efficient and robust search algorithms are needed to perform optimization [11, 13–15]. Due to the complexity of large-scale systems and uncertainties, it is difficult to develop accurate analytic models, explicitly formulate performance specifications, derive performance functionals, design optimal control laws, and synthesize hierarchical robust systems. Adaptive intelligence is defined as the ability of the closed-loop system to achieve the desired goals (for example, maximize mission effectiveness, agility, maneuverability, controllability, reliability, and survivability, while minimizing failures and losses) in an uncertain evolving environment through the system abilities to sense the environment, to learn and evolve, to adapt and perform reconfiguration to execute proper control actions. A block diagram of the proposed system architecture was illustrated in Fig. 3.

The multivariable input (control, reference, disturbances, noise)—output maps \(UY\) are used to study the dynamic system evolution [11]
\[
\dot{x}(t) = F(x, u, r, d, z), \quad y(t) = H(x), \quad x(t_0) = x_0
\]
which includes \(n\) subsystems, and the interconnected subsystems are modeled using nonlinear differential equations
\[
\dot{x}_{i(t)}(t) = F_{ij}(x, u_{i(t)}, r_{i(t)}, d_{i(t)}, z_{i(t)}),
\]
\[
y_{i(t)}(t) = H_{i(t)}(x), \quad x_{i(t)}(t_0) = x_{i(t_0)}
\]
where \(x \in X\) and \(x_{i(t)} \in X_{i(t)}\) are the system and subsystems state vectors; \(u \in U\) and \(u_{i(t)} \in U_{i(t)}\) are
the system and subsystems control vectors; $y \in Y$ and $y(i) \in Y(i)$ are the system and subsystems output vectors; $r \in R$ and $r(i) \in R(i)$ are the system and subsystems reference (command) vectors; $d \in D$ and $d(i) \in D(i)$ are the system and subsystems disturbance vectors; $z \in Z$ and $z(i) \in Z(i)$ are the system and subsystems uncertainties (environment changes, parameter variations, unmodeled dynamics, etc.); $F(x, u, r, d, z)$ and $F_{(i)}(x, u_{(i)}, r_{(i)}, d_{(i)}, z_{(i)})$ are the jointly continuous and Lipschitz fields.

The flight vehicle motion within the longitudinal and lateral axes is controlled by deflecting the control surfaces. The multivariable rigid-body dynamics is modeled using nine states ($v, a, q, \theta, \beta, p, r, \phi, \psi$), $x \in \mathbb{R}^9$. The pitch, roll and yaw moments are controlled by deflecting and changing the geometry of smart control surfaces. Using the tracking error vector, as given by $e(t) = Nr(t) - y(t)$, the block-diagram representation of the rigid-body vehicle with controller $u = \phi(e, x)$ is illustrated in Fig. 5 if the reference (command) surfaces. Using the tracking error vector, as given by $e(t) = Nr(t) - y(t)$, the block-diagram representation of the rigid-body vehicle with controller $u = \phi(e, x)$ is illustrated in Fig. 5.

It was emphasized that other requirements and specifications are imposed. Hierarchical distributed control systems for flight vehicles can be designed using the three-layer-architecture as documented in Fig. 3. Let us discuss the design of control algorithms for $j$-level layer of $k$-level hierarchical system. The controller for $j$th level of the $N$-layer hierarchical closed-loop system can be synthesized in the following form

$$u_j(t) = f_j \left( \mathbf{P}, \sum_{i=0}^{N} e_i(t), \sum_{i=0}^{N} x_i(t), \sum_{i=0}^{N} s_i(t), \sum_{i=0}^{N} p_i(t), p_j(t) \right)$$

where $u_j(t)$ is the control vector (output) for $j$th level; $f_j$ is the nonlinear map; $\mathbf{P}$ is the system performance variables, e.g., mission effectiveness, agility, maneuverability, controllability, reliability, survivability, failures, etc.; \( \sum_{i=0}^{N} e_i(t) \) is the error vector which represents the difference between the reference commands $r_j(t)$ and outputs $y_j(t)$; $\sum_{i=0}^{N} x_i(t)$ is the state vector; $\sum_{i=0}^{N} s_i(t)$ is the sensed information vector (inputs, outputs, state and decision variables, events, disturbances, noise, parameters, etc.) measured by $j$th and lower level sensors or estimated by the observer; $\sum_{i=0}^{N} p_i(t)$ is the parameter vector (time-varying flight vehicle aerodynamic coefficients, MEMS parameters, adjustable feedback gains changed through the adaptation, learning, evolution and reconfiguration); $p_j(t)$ is the parameter vector of the $j$th layer.

The simplest control algorithms are the proportional-integral-derivative (PID) controllers which can be designed using [11]

$$u(t) = \sum_{j=1}^{2N_p - 1} k_p(j-1) e^{2j-1}(t) + \sum_{j=1}^{2N_i - 1} k_i(j-1) e^{2j-1}(t)$$

$$+ \sum_{j=1}^{2N_d - 1} k_d(j-1) \frac{d^{2j-1}}{dt^{2j-1}} e(t)$$

where $N_p$, $N_i$, and $N_d$ are the positive integers assigned by the designer; $k_p(j-1)$, $k_i(j-1)$, and $k_d(j-1)$ are the proportional, integral, and derivative feedback coefficients.

For $N_p = 1$, $N_i = 1$ and $N_d = 1$, one has $u(t) = k_p e(t) + k_i \int e(t) dt + k_d \frac{d}{dt} e(t)$. The feedback gains $k_p$, $k_i$, and $k_d$ can be time-varying. In addition, $k_p$, $k_i$, and $k_d$ can be nonlinear functions of the state variables $x(t)$, tracking error $e(t)$, disturbances, etc.

It is obvious that the tuning problem cannot be performed through an empirical procedure, and the search for the feedback gains must be made through optimization in the performance domain, e.g., time domain cost functionals can be synthesized and used [11]. It must be emphasized that simple matching of the PID-type control laws to optimal controllers (synthesized using the performance functionals) can be achieved only for the centralized controllers synthesis. The performance functionals in the time domain can be given as

$$J(k_j) = \int_{t_0}^{t_f} \left[ f_p(\mathbf{P}) + \sum_{i=1}^{n} t e^{2i}(t, k_j) \right] dt,$$

$$J(k_j) = \int_{t_0}^{t_f} \left[ f_p(\mathbf{P}) + \sum_{i=1}^{n} t^2 | \dot{e}(t, k_j) | \right] dt,$$

$$J(k_j) = \int_{t_0}^{t_f} \left[ f_p(\mathbf{P}) + \sum_{i=1}^{n} | \dot{e}(t, k_j) | \right] dt,$$
where \( f_k(P) \) is the integrands which measure the systems and subsystem performance; \( k_j \) is the vector of the controller feedback coefficients which must be found to maximize the performance.

The functionals must integrate multi-dimensional performance objectives, such as, mission effectiveness, targeting, engagement, safety, agility, tracking, disturbance attenuation, robustness, etc. Using nonlinear functionals, the restricted structure consistent distributed control can be found for high-, medium-, and low-level layers, and the feedback gains are obtained solving the optimization problem.

Three-layer architecture of the hierarchical distributed flight management system has three levels. At the low level, the MEMS are controlled using the command signal and sensed data (for example, reference and actual displacements). The control signals are generated based on mathematical and logical (linguistic rules to describe the system operation) reasoning. The tasks that can be performed by a low-level layer are as follows:

1) “displace” (displacing the microstructure we change the geometry of control surfaces, and the pitch, roll and yaw moments result),
2) “compare the command and actual displacements” (obtain the tracking error),
3) eliminate the error between the reference and actual displacement (using the error, the corresponding pulsewidth modulated (PWM) signals are developed and feed to high-frequency transistors),
4) “measure the displacement,” “measure the developed force,” and “measure the aerodynamic load,”
5) “diagnose and detect failures” (for example, positive or negative values of the duty ratio ±1 correspond to counterclockwise or clockwise angular deflections, an increase of the duty ratio must lead to an increase of the current and electromagnetic field intensity, etc.).

Thus, we have a set of commands and events to attain the desired functionality. The low-level layer (with \( N \) ICs controllers to control \( N \) MEMS nodes) is primarily responsible for actuation, sensing, simple analysis, and control. The internal decision-making and local diagnostics can be performed at a low level. The medium-level layer, which controls smart flight surfaces (two flaps), develops the commands for the low-level layer in order to coordinate the actions of the multinode MEMS (actuator-sensor-ICs arrays). Flight vehicle control is performed by a supervisory flight management system, which, in addition to flight surface management, integrates many other functions (e.g., mission, path, position, propulsion, targeting, and engagement control), and the function/task data-intensive analysis is accomplished by the high-level layer. In particular, based upon the information obtained from the medium- and low-level layers, the high-level layer defines tasks (through adaptation, learning, evolution, reconfiguration, analysis, coordination, organization, decomposition, decision-making with performance analysis and outcome prediction, diagnostics, etc.) to attain the desired flying qualities, mission effectiveness, etc. The commands to displace the control surfaces are developed by the high-level layer based upon the overall analysis. It must be emphasized that the high-, medium- and low-level layers communicate with each other, and the high-level layer possesses a key role.

Adaptive control theory must be applied to develop and integrate key enabling methods, algorithms, and tools [11]. Knowledge-based adaptive systems must make optimal (robust) decisions based upon the intelligence and evolution strategies using specified requirements and priorities, monitoring (sensing) rapidly changing environment for entities of interest, recognizing those entities, inferring high-level attributes about those entities, etc. The adaptive systems use the data from different sensors, feedback commands (controls) are computed-generated-executed, and intelligent updates, evolution and reconfiguration are performed. The feedback for the sensor and control mechanisms are integrated, and particular emphasis is concentrated to gather the critical and essential data from the nodes of a greatest interest. Extensive information data sets must be constantly updated to guarantee a complete situational awareness, graduate evolution and intelligence using performance evaluation and state evolutions (qualitative and quantitative analysis), and ensure robust adaptation capabilities. To perform the inferences required, to develop an assessment of the current situation, and to predict performance and outcomes, extensive knowledge and information about the vehicle and environment are needed.

A multiple evolutionary learning concept through robust adaptation is implemented to design hierarchical distributed closed-loop systems with high-, medium-, and low-level layers. The MEMS nodes exhibit complex behavior which can be optimized using low-level evolutionary adaptation using reconfiguring-while-learning algorithms. The low-level layer performs the following major local functions: sensing, actuation, adaptation, reconfiguration, recognition, diagnostics, simple assessment and prediction. Reinforcement learning is performed based upon the prioritized objectives through upper level layers. The multinode MEMS (actuator-sensor-ICs arrays) behavior and performance are analyzed by the medium- and high-level layers to collect and assess the evidence data. Decision trees are used to provide a comprehensive set of strategies and algorithms, simplify and improve (optimize) them, attain robustness and comprehensibility, and achieve the adaptation.
VII. IMPLEMENTATION ISSUES AND SIMULATION RESULTS

Analysis, control, and simulations are examined for a micro air vehicle with 20 cm span. The MATLAB environment is used, and complementary highly efficient m-files are developed and tested. Hierarchical distributed closed-loop system for a flight vehicle is designed. Different architectures were examined, and the number of layers is based upon the complexity of systems to be controlled, e.g., the number of nodes, objectives, tasks, environment, etc. The hierarchy level is defined by hardware and software complexity (rate of tasks completion, rate of continuous and discrete events, bandwidth, sampling time, update rate, etc.), as well as by the overall specifications and requirements imposed. Complex problems and tasks were logically decomposed into simpler subproblems and subtasks which are easy to understand, support, and implement using the state table of rules. For example, for each rule, the actuation and sensing actions of the MEMS node are determined using the system state, event, decision, and performance variables. The subtasks must be performed in the defined sequence scenarios that lead to the desired operation, and the system architecture was synthesized taking into account the above mentioned issues. Control of cooperative multinode MEMS is performed at the lower level of the hierarchy, and the higher level layers are defined based upon the overall objectives, analytical and numerical complexity of problems/tasks, information flow, etc. Thus, the complexity gradually rises from single-input/single-output MEMS node to adaptive control of smart flight surfaces and air vehicle. For a three-layer configuration studied, the possible architecture consists of the following:

1) high-level layer of vehicle flight management: flight vehicle control with intelligence, performance analysis, outcome prediction, evolutionary leaning, adaptation, and reconfiguration,

2) medium-level layer: adaptive coordinated and autonomous reconfigurable control of smart flight surfaces (flight surfaces can integrate hundreds of MEMS nodes which have sensors, actuators and ICs),

3) low-level layer: simple feedback and sensing for MEMS nodes.

Thus, the problems are decomposed into subproblems performed by different layers (which can operate at different sampling rates) with the synthesized layouts of decomposed tasks and functions. In particular, the problems are decomposed by the high-level layer into narrow tasks and functions which are fed (with or without adaptation, analysis, diagnostics, estimation, implementation, performance, realization, as well as other details) to the medium-level layer, which further decomposes the tasks and supervises the low-level layer. This hierarchically distributed closed-loop system architecture was shown in Fig. 3. Different operating systems, interfaces, and platforms are supported by the existing high-performance software. There is a critical need for novel efficient algorithms to implement closed-loop systems for flight vehicles. Using Java, the designer lays out and supports the hierarchical control (in if-then-else execution format), generates codes for different platforms, adds and removes modules, sets up communication and networks based upon timing requirements, writes data to the shared memory buffers and reads data from the buffers, develops protocols, codes and encodes data from the buffers using ASCII and Xdr file formats, performs diagnostics, etc. Intelligence, evolution, coordination and autonomy through task generation, organization and decomposition, adaptation with performance analysis and outcome prediction, diagnostics, estimation, reconfiguration, fault tolerance, robustness, and other functions are performed through sensing-actuation, learning, evolution, analysis, evaluation, behavioral (dynamic and steady-state performance) and task optimization. Hierarchical algorithms are synthesized based upon the system complexity, and the state-of-the-art hardware and software must be implemented. The analysis of complexity, hierarchy, data flow, and module design, allows the designer to synthesize the system architectures starting from lowest structural level and then governing and augmenting the lower levels to upper levels based on physical relationships, functional correlation, order, sequence, and arrays to attain the desired level of performance, capabilities, robustness, adaptability and efficiency. It should be emphasized that the neural network concept was used to attain the optimization and reconfigure (adjust) the feedback gains [11, 14, 15] performing parametric optimization. In this way, the adaptive controller is found.

The longitudinal-lateral maneuvering of micro air vehicle is studied. The deflection of $2 \times 2$ cm (in the $xy$ plane) smart control surface with 81 MEMS nodes ($9 \times 9$ array) for two distinct times (0.1 and 0.2 s) during the vehicle maneuvering (0.2, 0.4 and 1.25 rad turn) is illustrated in Fig. 6 (the scale for the $z$-axis is mm). As illustrated in [9], the microscale motion devices guarantee microsecond settling time, and using these MEMS capabilities, active aerodynamic flow control can be achieved.

The output dynamics of the micro air vehicle is documented in Fig. 7. In particular, we assign the following commands (desired Euler angles) 0.2, 0.4, and 1.25 rad (the initial values are zero). The first figure illustrates three resulting plots. The settling times are 3.2, 1.8, and 6.9 s for the elevation, bank and azimuth angles. It should be emphasized that the overall rigid-body vehicle dynamics is primarily...
a function of the six-degree-of-freedom vehicle model, and the MEMS actuators transient are in the microsecond range. For the considered micro air vehicle, relatively small deflections are achieved using the translational MEMS actuators (see Fig. 6), and the vehicle velocity is low. The second figure illustrates the Euler angles evolution if the command elevation, bank and azimuth angles are 0.2, 0, and 1.25 rad (initial values for angles are 0, 0.4, and 0 rad). The application of smart control surfaces improves the vehicle performance compared with conventional actuators. Therefore, the advantages of the MEMS actuators make the developed results very attractive and promising.

VIII. CONCLUSIONS

In this article, smart flight surfaces were designed using the MEMS arrays controlled by the synthesized hierarchical distributed system. The distributed control design paradigm, which results in adaptive control algorithms, guarantees the desired performance specifications. The performance characteristics were assessed performing analysis, modeling, and simulations. It was illustrated that the adaptive aerodynamic control can be performed using MEMS actuators-sensors-ICs arrays. The hierarchical adaptive control system attains the spectrum of objectives, specifications, and requirements imposed. The desired pitch, roll, and yaw moments (which are nonlinear functions of control surfaces deflection and geometry, vehicle aerodynamic characteristics, velocity and altitude) are produced to achieve the desired performance and specifications. The application of smart flight surfaces significantly improves flying qualities in the full operating envelope, increases mission effectiveness, enhances agility, controllability, maneuverability, leads to safety and survivability, extends cruise range, reduces fuel consumption, etc. Therefore, the reported concept can be applied and implemented for the next generation of munitions, missiles, unmanned, and manned flight vehicles.

REFERENCES

Sergey Edward Lyshevski was born in Kiev, Ukraine in 1957. He received the M.S. (1980) and Ph.D. (1987) degrees from Kiev Polytechnic Institute, all in electrical engineering.

From 1980 to 1989 he held faculty positions at the Department of Electrical Engineering at Kiev Polytechnic Institute. From 1989 to 1992 he was the Microelectronic and Electromechanical Systems Division Head at the Academy of Sciences of the Ukraine. Since 1993 he has been with Purdue University at Indianapolis, where he is currently an Associate Professor in the Department of Electrical and Computer Engineering. His current teaching and research interests include nonlinear system theory, optimization, MEMS and NEMS system integration, and CAD.

He is the author of more than 250 technical papers and 8 books.