Resilient-based Control Reconfiguration of Autonomous Systems

Sehwan Oh[1], Benjamin Lee[2], Michael Balchanos[3], Dimitri Mavris[4], and George Vachtsevanos[5]

[1,2,3,4,5]*Georgia Institute of Technology, Atlanta, GA, 30024, United States of America*

soh48@gatech.edu
blee78@gatech.edu
michael.balchanos@asdl.gatech.edu
dimitri.mavris@aerospace.gatech.edu
gjv@ece.gatech.edu

ABSTRACT

This paper introduces a design methodology for resilient-based control reconfiguration of Unmanned Autonomous Systems (UAS) when extreme disturbances, such as a largely growing fault or component failure mode occur. It is documented that more than 40% of Class air mishaps are attributed to Unmanned Aerial Vehicles (UAVs). There is an urgent need to improve the operational integrity, resilience and reliability of such critical assets. An optimal control approach with Differential Dynamic Programming (DDP) and Model Predictive Control (MPC) is introduced in this paper as a means for control authority redistribution and reconfiguration; therefore, the system continues performing its mission while compensating for the impact of the extreme disturbances. Prognostic knowledge is considered in a quadratic cost function of the optimal control problem as a soft constraint. A trade-off parameter is introduced between the prognostic constraint and the terminal cost. An autonomous ground operable under-actuated hovercraft is used to demonstrate the efficacy of the proposed reconfiguration strategy, and it is extendable to other cyber physical systems.

1. INTRODUCTION AND BACKGROUND

UAS, such as unmanned aerial, ground, surface, and underwater vehicles, are being extensively utilized in research and application domains exploring potential uses and developing new ones. One of the primary concerns of practical UAS utilization in real operations is safety (Downes, 2015). In many applications, human operators are still necessary to help address unforeseen and extreme disturbances, such as component faults that lead to failure. The motivation for this research is founded upon the idea of system resilience for system safety and reliability under extreme disturbances. System resilience is defined as:

Definition [Resilience]: The intrinsic ability of a system to adjust its functioning prior to, during, or following changes and disturbances, and thus, to sustain required operations even after a major mishap or in the presence of continuous stress (Hollnagel, Woods, and Leveson, 2007).

Balchanos (2012) thoroughly reviewed various resilience-related research findings and addressed an assessment method of complex dynamic system resilience, which embraces system capability. Tran (2016) also suggested a resilience assessment method based on time-dependent system reliability by using a probabilistic measure.

According to the definition of resilience, situational awareness, prediction, planning, and action are necessary capabilities for a resilient system. On one hand, fault diagnosis and failure prognosis aim to detect accurately with specified false alarm rate the initiation of a fault while prognosis attempts to predict the system's Remaining Useful Life (RUL). (Vachtsevanos, Lewis, Roemer, Hess, and Wu, 2006). A particle filtering-based diagnosis and prognosis framework has been highlighted as a popular approach (Brown, Bole, and Vachtsevanos, 2010), and its details can be found in Orchard's dissertation (2006).

On the other hand, proper adjustments to control actions can assure resilient behaviors. Specifically, for a fault and failure in a critical component, a Fault Tolerant Control System (FTCS) has been researched for decades. It was motivated by commercial aircraft accidents (Zhang & Ziang, 2008). Zhang and Ziang (2008). Clements (2003) developed a hierarchical control architecture showing the interconnections among fault detection & identification, set-point controller, control redistribution, control gain adaptation, and component restructuring. Ge, Kacprzynski, Roemer, and Vachtsevanos (2004) introduced a higher level of adaptive system framework by using an Automated Contingency Management (ACM) concept. Drozeski, Saha, and Vachtsevanos (2005) proposed a three-tier hierarchical control scheme as Active FTCS. Tang, Kacprzynski, Goebel, Saxena, Saha, and Vachtsevanos (2008) extended the ACM framework by integrating it with a prognostics module. Tang, Hettler, Zhang, and DeCastro (2011) extended further and tested a Prognostics and Health Management (PHM)

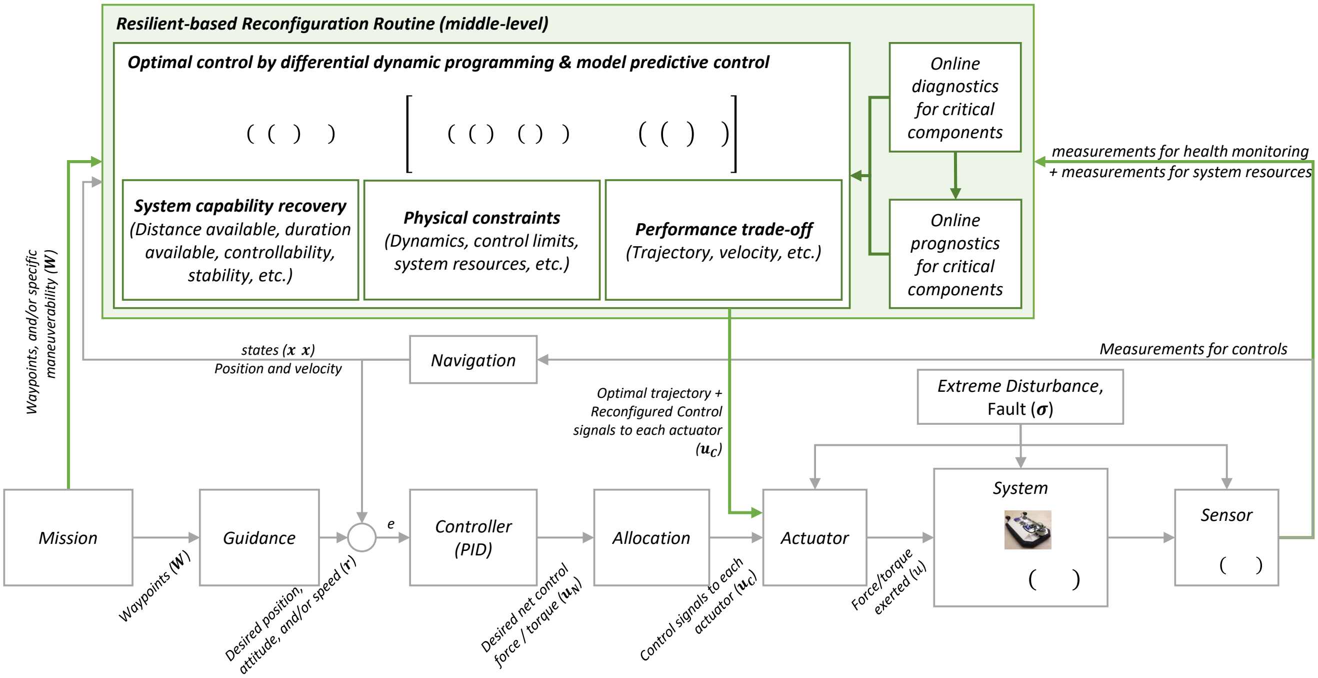

Resilient-based Reconfiguration Routine (middle-level)
Optimal control by differential dynamic programming & model predictive control
(( ))
(( ) ( )) (( ))
System capability recovery
(Distance available, duration available, controllability, stability, etc.)
Physical constraints
(Dynamics, control limits, system resources, etc.)
Performance trade-off
(Trajectory, velocity, etc.)
Online diagnostics for critical components
Online prognostics for critical components
measurements for health monitoring
+ measurements for system resources
Waypoints, and/or specific maneuverability ($\boldsymbol{W}$)
states ($\boldsymbol{x}$ $\boldsymbol{x}$)
Position and velocity
Navigation
Measurements for controls
Optimal trajectory + Reconfigured Control signals to each actuator ($\boldsymbol{u}_C$)
Extreme Disturbance, Fault ($\boldsymbol{\sigma}$)
Mission
Guidance
e
Controller (PID)
Allocation
Actuator
System
( )
Sensor
( )
Waypoints ($\boldsymbol{W}$)
Desired position, attitude, and/or speed ($\boldsymbol{r}$)
Desired net control force / torque ($\boldsymbol{u}_N$)
Control signals to each actuator ($\boldsymbol{u}_C$)
Force/torque exerted ($\boldsymbol{u}$)

Figure 1. Overview of the proposed methodology on a closed-loop control schematics.

enhanced ACM framework with a real-time mobile robot test application. Brown, Georgoulas, Bole, Pei, Orchard, Tang, Saha, Saxena, Goebel, and Vachtsevanos (2009) proposed prognostics enhanced low-level reconfigurable control for an electrical component. Bole, Tang, Goebel, and Vachtsevanos (2011) described a fault adaptive control architecture, and Bole (2013) addressed uncertainties in prognostics and reconfigurable control allocation strategies.

As a theoretical foundation, this study takes the hierarchical reconfiguration architecture developed by Drozeski et al. (2005), and focuses on the middle-level reconfiguration module. The following sections of the paper cover the proposed resilient-based reconfiguration strategy in Section 2, the proof of concept using an under-actuated hovercraft example in Section 3, and summary of the contribution and future work in Section 4.

2. TECHNICAL APPROACH

The main contribution of this research effort is a middle-level reconfigurable control framework, as shown in Figure 1. The configuration borrows from the comprehensive three-level reconfiguration architecture proposed by Drozeski et al. (2005) and Brown et al. (2009), as illustrated in Figure 2.

Under small disturbances, the low-level reconfiguration module compensates for the effects of small disturbances by adjusting set points for the actuator components. In the case of extreme disturbances, however, due to the severe degradation of the system capability, the system is not able to satisfy mission requirements; thus, it cannot perform / complete the given mission. To resolve this issue, the middle-level reconfiguration module is introduced to extend and recover the system capability by reconfiguring the guidance and control strategy. The system capability recovery in terms of an extended RUL is achieved at the expense of degraded system performance. It is noted that the proposed reconfiguration strategy does not entail any hardware changes (replacement or insertion of new hardware). It addresses only the software components of the framework. Therefore, the trade-off between the performance requirements and increased RUL must be carried out properly in the middle-level reconfiguration module. This trade-off is the essence of the middle-level reconfiguration.

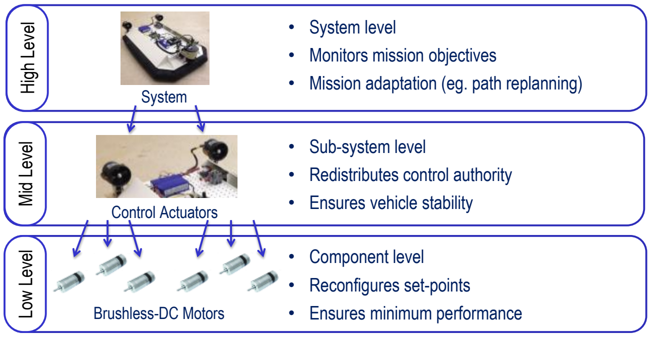

High Level
System
• System level
• Monitors mission objectives
• Mission adaptation (eg. path replanning)
Mid Level
Control Actuators
• Sub-system level
• Redistributes control authority
• Ensures vehicle stability
Low Level
Brushless-DC Motors
• Component level
• Reconfigures set-points
• Ensures minimum performance

Figure 2. Reconfigurable control architecture with three tier strategies (modified and redrawn based on Figure 3 in Brown et al., 2009)

2.1. Requirements

The middle-level reconfiguration strategy aims to extend the system capability (Time to Failure) by trading off performance attributes. The strategy involves only the system's software assuming that the hardware complement remains as designed. Then, the recovered system capability has to satisfy mission requirements. In addition, the middle-level reconfiguration can be carried out properly only if a system is controllable under extreme disturbances. Dorf and Bishop (1998) defined the system controllability requirements.

2.2. Resilient-based Reconfiguration Strategy

The middle-level consists of system-level Guidance, Navigation, and Control (GNC) modules. The guidance and control laws have a major impact on generating desired control forces and/or torques. Once an ongoing disturbance is diagnosed as being extreme – thus, the low-level reconfiguration is not sufficient to achieve stated mission objectives – the middle-level reconfiguration routine is activated. This paper assumes that the diagnostics and prognostics information, as well as the decision from the low-level reconfigurable controller, are given.

As a means for control reconfiguration, optimal control theory is a powerful tool for deriving control policies, and it is the backbone of the reconfiguration strategy in this study. DDP, developed by Jacobson and Mayne (1970), is used to solve a general nonlinear optimization problem. The key attribute of DDP is the fact that it performs simultaneously trajectory optimization and control signal generation. DDP is a finite time horizon control algorithm; thus, it is natural to borrow and apply a MPC method until the system meets the success criteria of a given mission.

Figure 3 represents the schematics of the middle-level reconfigurable control framework. If the system is controllable under an extreme disturbance, the finite-time optimal control law generates and applies the first input sequence; it is iterated next with the updated states until a given target is met. System performance degradation is inevitable as the recovery actions compensate for such extreme disturbances. Therefore, system performance must be evaluated and compared with the required performance criteria. If the reconfiguration level does not meet the specified performance criteria, the top-level of the control hierarchy is activated to achieve mission adaptation.

2.2.1. Formulation

A general optimal control problem is stated as: find the optimal input, $\boldsymbol{u}^*$, minimizing a nonlinear cost functional as follows:

$$^{*} \quad \arg\min f(\boldsymbol{x}\ \ \boldsymbol{u}) \tag{1}$$

where $\boldsymbol{u} \in \mathfrak{R}^n$ is a control input (desired force/torque) vector, $\boldsymbol{x} \in \mathfrak{R}^m$ is a state vector, and $f(\cdot)$ is a nonlinear cost function. A finite time horizon optimal control problem is formulated as:

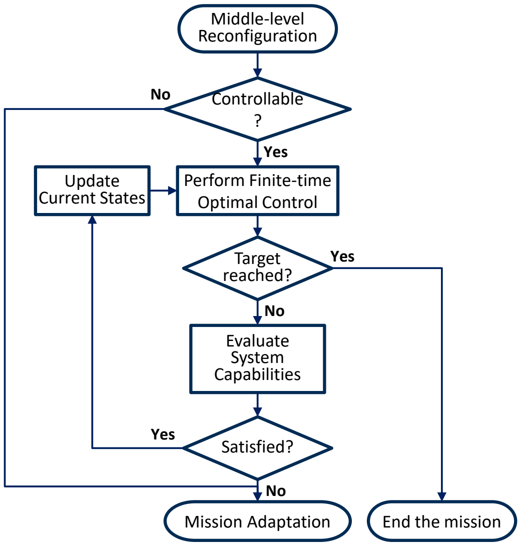


Figure 3. Flowchart of the proposed control reconfiguration.

$$(\boldsymbol{x}(t_0), t_0)$$

$$\left[\int \quad (\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau)\, d\tau \qquad (\boldsymbol{x}(t_f), t_f)\right] \tag{2}$$

subject to:

$$\begin{gathered} — \quad F(\boldsymbol{x}(t), \boldsymbol{u}(t)) \\ (\boldsymbol{x}(t), \boldsymbol{u}(t)) \le 0 \end{gathered} \tag{3}$$

where t_0 is an initial time and t_f is the terminal time. $l(\cdot)$ is a scalar running cost, $\Phi(\cdot)$ is a scalar terminal cost, $F(\boldsymbol{x}(t), \boldsymbol{u}(t))$ represents the nonlinear system dynamics as an equality constraint for the optimization problem, and $g(\boldsymbol{x}(t), \boldsymbol{u}(t))$ is a general function for inequality constraints. This optimal control problem can be solved by a DDP approach. By exploiting Bellman's principle, the cost values are propagated backward in time if we know $V(\boldsymbol{x}(t_f), t_f)$ (Bellman, 1957). Bellman's principle in a discrete-time domain is expressed in Equation (4).

$$(\boldsymbol{x}(t_k), t_k)$$

$$_{\boldsymbol{u}(t_k)}[l(\boldsymbol{x}(t_k), \boldsymbol{u}(t_k), t_k)\Delta t \qquad (\boldsymbol{x}(t_{k+1}), t_{k+1})] \tag{4}$$

Cost functions are usually formulated as quadratic functions of states and control inputs, as shown in Equations (5) and (6).

$$(\boldsymbol{x}(t_f), t_f) \quad \frac{1}{2}(\boldsymbol{x}(t_f) - \boldsymbol{r})^T K_f (\boldsymbol{x}(t_f) - \boldsymbol{r}) \tag{5}$$

where $\boldsymbol{r}$ is a target state vector, and K_f is a weighting matrix. It represents the weighted energy of a terminal state error, expressed as the difference from a given target.

$$(\boldsymbol{x}(t_k),\boldsymbol{u}(t_k),t_k)$$
$$\frac{1}{2}\quad(t_k)^T K\boldsymbol{x}(t_k)\quad\frac{1}{2}\quad(t_k)^T R\boldsymbol{u}(t_k) \tag{6}$$

where K and R are weighting matrices. The right hand-side of Equation (6) refers to the energy of the states and control inputs, respectively, at each time instance.

As a constraint, the system capability must be sufficient to meet stated mission requirements.

$$\text{system capability} \geq \text{requirements} \tag{7}$$

The qualitative constraints must be expressed as measurable quantities. Candidates are energy, controllability, maneuverability, stability, etc. By the assumption that system capability is related to system control authority, the inequality constraint can be expressed as:

$$\boldsymbol{\sigma}(t_k) > \boldsymbol{\sigma}_{\text{req.}} \tag{8}$$

subject to:

$$\dot{\boldsymbol{\sigma}}(t_k)\qquad(\boldsymbol{x}(t_k),\boldsymbol{u}(t_k),t_k) \tag{9}$$

where $\boldsymbol{\sigma} \in \Re^{m_2}$ is the system capability-relevant state vector, and $\boldsymbol{\sigma}_{\text{req.}}$ is the state vector for mission requirements. $\boldsymbol{\sigma}(t_k)$ is modeled as a function of $\boldsymbol{u}(t_k)$ and $\boldsymbol{x}(t_k)$, as in Equation (9). In general, hard constraints in states, specifically at a terminal time, are difficult to handle in MPC. An alternate approach is to set them as soft constraints by integrating them into a cost function. Then, Equations (5) and (6) can be rewritten as Equations (10) and (11):

$$\begin{aligned}&(\boldsymbol{x}(t_f),t_f)\\&\qquad\frac{1}{2}\left(\boldsymbol{x}(t_f)-\boldsymbol{r}\right)^T K_f\left(\boldsymbol{x}(t_f)-\boldsymbol{r}\right)\\&\qquad\quad+\rho\quad\frac{1}{2}\quad(t_f)^T K_f^R\boldsymbol{\sigma}(t_f)\end{aligned} \tag{10}$$

$$\begin{aligned}&(\boldsymbol{x}(t_k),\boldsymbol{u}(t_k),t_k)\\&\qquad\frac{1}{2}\quad(t_k)^T K\boldsymbol{x}(t_k)\quad\frac{1}{2}\quad(t_k)^T R\boldsymbol{u}(t_k)\\&\qquad\quad+\rho_R\frac{1}{2}\quad(t_k)^T K^R\boldsymbol{\sigma}(t_k)\end{aligned} \tag{11}$$

where K_f^R & K^R are weighting matrices, and ρ_f & ρ_R are reconfiguration parameters. In the case that system controllability is satisfied even after the component failure, ρ_R will be a main driver for the trade-off. Now, we can concatenate the state variable vectors into one long state vector and several weighting matrices into two large matrices:

$$[\boldsymbol{x}\quad\boldsymbol{\sigma}]^T \tag{12}$$

$$K\quad\begin{bmatrix}K & \mathbf{0}\\ \mathbf{0} & \rho_f K_f^R\end{bmatrix}$$
$$K=\begin{bmatrix}K & \mathbf{0}\\ \mathbf{0} & \rho_R K^R\end{bmatrix}$$

where the new state vector is $\boldsymbol{x} \in \Re^{m+m_2}$. Then, Equations (10) and (11) are reorganized as Equations (5) and (6). Determining the weighting matrices and importance parameters is heavily dependent on the controller designer's experience. It is noted though that what is affecting the optimal control performance are ratios between weighting parameters. As a starting point, it behooves to normalize them with reference values (e.g., maximum) of each variable.

2.2.2. Stability Analysis

Let us rewrite Equation (2) as:

$$\begin{aligned}&V_N(x(t_0),\boldsymbol{u})\\&\qquad\sum_{j=0}^{N-1}\left(x(t_j),u(t_j)\right)\Delta t\qquad\left(x(t_N)\right)\end{aligned} \tag{13}$$

where $x(t_0)$ is the current state, is the control input sequence, and $\left(x(t_N)\right)$ is the terminal cost. The cost function $V_N(x(t_0),\boldsymbol{u})$ evaluates costs for N discrete time segments at $x(t_0)$. Define $\boldsymbol{u}^*\left(x(t_0)\right)$ the optimal control input sequence given the initial condition, $x(t_0)$, and $V_N^*(\cdot)$ the costs along the optimal control input, $\boldsymbol{u}^*(\cdot)$, at any given current state. Considering discrete system dynamic constraints,

$$x(t_{k+1})\qquad\left(x(t_k),u(t_k)\right) \tag{14}$$

where $u(t_k) \in \mathbb{U}$, and $\mathbb{U}$ is a feasible input set, $V_N^*(\cdot)$ can be considered as a Lyapunov function if,

$${}_N^*\left(x(t_{k+1})\right)-V_N^*\left(x(t_k)\right)\leq 0 \tag{15}$$

for all t_k. Then, the MPC, with the cost function expressed as Equation (13), is stable. The proof is as follows:

Suppose that there exists an optimal control input, $\boldsymbol{u}^*(t_k)$ $[u^*(t_k)\quad u^*(t_{k+1})\quad\cdots\quad{}^*(t_{k+N})]$, and the corresponding state, $\boldsymbol{x}^*\left(x(t_k)\right)$, sequences at t_k. Now, consider the control input sequence at time t_{k+1} as:

$$\begin{aligned}&\left(x(t_{k+1})\right)\\&\qquad[u^*(t_{k+1})\quad\cdots\quad{}^*(t_{k+N})\quad u(t_{k+N+1})]\end{aligned} \tag{16}$$

Equation (16) is not an optimal control sequence because $u(t_{k+N+1}) \in \mathbb{U}$ is not optimal. By the definition of the cost function,

$$\begin{aligned}&\left(x(t_{k+N+1})\right)\quad\left(x(t_{k+N}),u(t_{k+N})\right)\Delta t\\&\qquad\qquad\leq V_f^*\left(x^*(t_{k+N})\right)\end{aligned} \tag{17}$$

Therefore,

$$
\begin{aligned}
{}_N\left(x(t_{k+1}),\quad \left(x(t_{k+1})\right)\right)& \\
\le V_N^*\left(x^*(t_k)\right) - \quad &\left(x(t_k),u^*(t_k)\right)\Delta
\end{aligned} \tag{18}
$$

Finally,

$$
\begin{aligned}
{}_N^*\left(x^*(t_{k+1})\right) \le V_N\left(x(t_{k+1}),\quad \left(x(t_{k+1})\right)\right)& \\
\le V_N^*\left(x^*(t_k)\right) - \quad &\left(x(t_k),u^*(t_k)\right)\Delta
\end{aligned} \tag{19}
$$

Proof of Equation (15) is found in (Pannocchia, Rawlings, and Wright, 2011).

3. THE EXPERIMENTAL CONFIGURATION

The proposed design methodology of the middle-level reconfigurable control has potentials to cyber physical systems, which entail software and hardware integration, in that the reconfiguration manages software components. To demonstrate its efficacy, an autonomously operable under-actuated hovercraft was used as a testbed (Kim et al., 2013; Sconyers et al., 2013). The hovercraft dynamics model was derived on the basis of a ground-fixed coordinate system, as depicted in the right side of Figure 4. The hovercraft operates with two differential thrust fans with electrical motors and a LIDAR sensor for simultaneous localization and mapping.

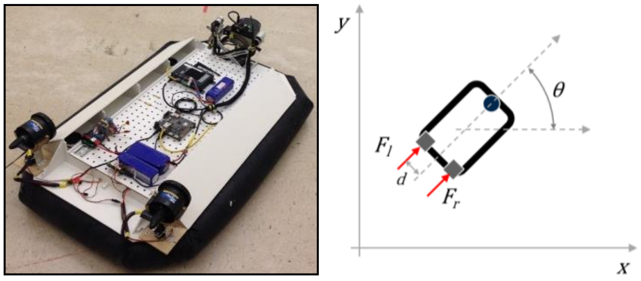


Figure 4. The autonomously operable hovercraft with two differential thrusts (left), and 2D hovercraft dynamics and kinematics representation (right).

3.1. Hovercraft Dynamics Model

The hovercraft is assumed to move in two-dimensional planar motion; thus, it is an under-actuated system given two input controls. Equations (20) are the system dynamics model; x and y are absolute positions on the ground fixed coordinate, θ is a heading angle, $\dot{X}$ is a velocity, $\ddot{X}$ is an acceleration, m is the mass, J is the moment of inertia of the hovercraft, d is the distance between a thruster and an imaginary longitudinal line crossing the mass center while assuming that the mass center coincides with the geometric center, and F_l & F_r are left and right thrust forces, respectively. Based on the system dynamics equations, the state is $\{x \quad ,\theta, \qquad \dot{\theta}\}^T$, and the input is $\{F_l, F_r\}^T$. Han, and Zhao (2004) evaluated the underactuated hovercraft controllability. The analysis showed that the existence of the yaw torque can guarantee the system controllability. It implies that one thrust motor failure does not affect the controllability as long as the other motor can produce proper torque values.

Table 1 shows the system properties used for the following experiments.

$$
\begin{aligned}
\ddot{x} &\quad - \text{—} \qquad F_l \cdot \cos\theta + F_r \cdot \cos\theta \\
\ddot{y} &\quad - \text{—} \qquad F_l \cdot \sin\theta + F_r \cdot \sin\theta \\
\ddot{\theta} &\quad -\frac{r}{J}\dot{\theta} \qquad (F_r - F_l)
\end{aligned} \tag{20}
$$

Table 1. System properties.

Parameters	Values	Description
m (kg)	11.8	Vehicle mass
J $(kg \cdot m^2)$	1	Moment of Inertia
d (m)	0.25	Moment arm
d_t (-)	0.05	Frictional damping (translation)
d_r (-)	0.005	Frictional damping (rotation)
F_{max} (N)	2	Control input constraint (max.)
F_{min} (N)	-2	Control input constraint (min.)

3.2. Fault Growth Model

A fault growth dynamics model is given as a function of time and actuator control inputs as:

$$
\dot{\sigma}(t) = \rho_\sigma \cdot u_{.}^{2} \qquad \sigma \qquad \omega_\sigma(t) \tag{21}
$$

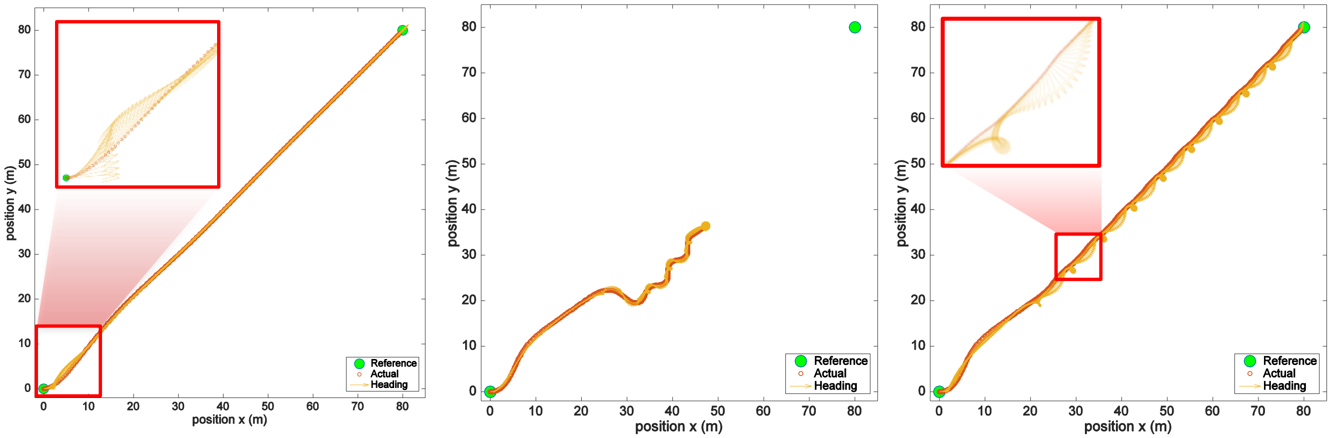
where σ is the state of a fault on the right thrust motor, $\omega_\sigma(t)$ is noise, ρ_σ is a coefficient representing the fault growth rate with respect to an actuator control input, and σ_0 is a control-independent parameter. As a dimensionless representation, the fault severity is ranked from 1 to 10 with one as a healthy condition and 10 as an indication of a component failure. At the fault severity 10, the motor control thrust force is no longer active. For simplicity, the impact of the fault mode on the effective thrust force is assumed inversely proportional to the severity of the fault, as shown in Equation (22).

Table 2 shows the fault growth model parameters that were used in the following tests.

$$
F_{\text{actual}} \qquad {}^{F_{\text{desired}}}\!/_{\sigma} \tag{22}
$$

Table 2. Parameters for the fault growth model.

Parameters	Values	Description
ρ_σ	2.5	Control input effect coef.
σ	0.02	Operational time effect coef.



(a) (b) (c)

Figure 5. Hovercraft position trajectory result comparisons: (a) healthy condition; (b) nominal controller under faulty condition; and (c) reconfigurable controller under faulty condition.

3.3. Energy Consumption Model

The system capability in this example is expressed as the traveling distance for the vehicle starting at the origin and terminating at a goal point, and it is closely tied to the system energy and the energy consumption model. The electric energy consumption rate is modeled as a quadratic function of control inputs.

$$\dot{e}(t) = \rho_e \cdot \boldsymbol{u}(t)^T \boldsymbol{u}(t) + \omega_e(t) \tag{23}$$

where $e(t)$ is the cumulative amount of energy consumed, $\dot{e}(t)$ is the energy consumption rate, and $\omega_e(t)$ is noise. Notionally, the maximum energy available is set to 300 in the example (dimensionless). If the total consumption reaches its maximum value, it is impossible to move the hovercraft any longer; thus, it becomes uncontrollable. For the experiment, ρ_e was set to 10.

3.4. Nominal GNC

A nominal-phase GNC module consists of the line-of-sight (LOS) guidance law and the dynamic inversion nonlinear controller addressed by Kim et al. (2013). The LOS guidance law forces the system to reduce the errors in the heading angle and the shortest distance between the current position and a trajectory path, at each control time instant. The controller controls the desired surge velocity and the heading angle.

4. SIMULATION RESULTS AND DISCUSSION

The hovercraft test mission is to move from a starting point, (0, 0), to a target point, (80, 80). A fault occurs in the right thrust motor during the operation initiated at 50 sec., and its severity monotonically increases as modeled in Equation (21). Figure 5 depicts the hovercraft position and heading trajectories (a) under healthy condition, (b) with the nominal controller under faulty condition, and (c) with the proposed reconfigurable controller under faulty condition. In the reconfigurable controller, ρ_f and ρ_R were set to 100. As expected, the nominal controller could not handle such an extreme fault and could not reach the target point at the end. With the reconfigurable controller, on the contrary, the hovercraft reached the target, but it exhibited an oscillatory behavior in the middle of the operation. This behavior is attributed to the redistributed control authority. As illustrated in Figure 6, the reconfigurable controller endowed agility characteristics to the healthy thrust motor while suppressing the usage of the faulty one. At $t_{k+\delta}$, the actual force exerted from the right faulty thrustor was less than the left healthy thrustor. Instead of exerting more effort on the faulty thrustor, the controller forced to turn the vehicle right until the heading angle pointed backward, and then produced a reverse thrust on the left motor to turn the vehicle heading back to the forward direction as well as to proceed in the direction of the target point. This control strategy repeated until the hovercraft reached the given target.

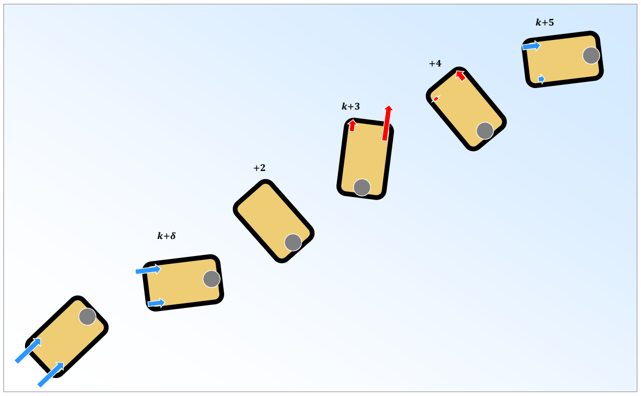


Figure 6. Pictorial representation of the redistributed optimal control sequence.

Through reconfiguration, system capability was recovered from the impaired condition, although the system performance was degraded. Figure 7 compares the system performance between normal and faulty conditions. After 50 seconds, as a critical fault mode was initiated and evolved, the speed to get to the target decreased; thus, the mission completion time was almost doubled compared to the healthy case. Energy consumption rate decreased after the fault mode, but due to the longer operational time, total energy consumption was slightly greater than the normal case. The trade-offs are obvious, as suggested previously.

Table 3 shows the comparisons of system capabilities in maximum traveled distances available and corresponding traveling time. As an extreme fault mode occurs, system capabilities by the nominal controller dramatically decreased to less than one third of the maximum traveling distance available. This was mainly due to an excessive usage of the faulty motor as the nominal controller attempted to maintain its heading angle; thus, the energy consumption was expedited and the component failed quickly. After reconfiguration, the maximum traveling distance recovered although not fully. To extend the distance, the mission was completed at a slower rate. Table 4 shows the impact of the reconfiguration parameters. The larger the values, the more penalties are assigned to the cost function. Finally, the maximum computational time for a single control input was 0.16 seconds on a regular desktop computer. It demonstrates the potential for the practical applicability of the proposed reconfiguration framework.

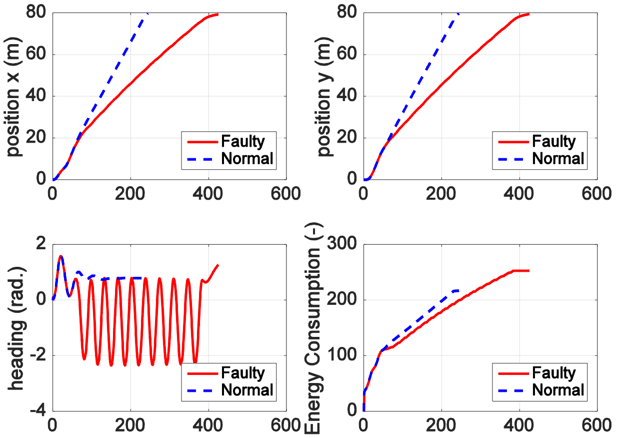


Figure 7. Solid red is reconfigured trajectory under faulty condition; and dashed blue is normal trajectory.

Table 3. Comparisons of system capabilities.

Capability	Healthy	Faulty	
		No Reconf.	Reconf.
Max. distance (m)	177.61	55.87	143.41
Time to mission complete (sec.)	241	Incomplete	422
Time to the component failure (sec.)	-	75.5	480.5

Table 4. Impact of the reconfiguration parameter.

Capability	Reconfiguration Parameters		
	0	100	200
Max. distance (m)	103.50	143.41	194.38
Time to mission complete (sec.)	Incomplete	422	527
Time to the component failure (sec.)	No failure	480.5	482.5

5. CONCLUSION

The middle-level reconfigurable control and control authority redistribution framework introduced in this paper demonstrated its essential capability to improve system resiliency by handling an extreme disturbance properly. The proposed framework was tested via the under actuated hovercraft simulation. The results demonstrated the efficacy of the approach. Future work is intended to address uncertainties, which may degrade the performance of the reconfiguration strategy. Also, adaptive mechanisms for systems to autonomously manage the level of reconfiguration online will help systems to be more resilient. Furthermore, integration of the low and high-level reconfiguration policies will improve the overall system performance.

REFERENCES

Balchanos, M. G., (2012). *A probabilistic technique for the assessment of complex dynamic system resilience*. Doctoral dissertation. Georgia Institute of Technology, Atlanta, USA.

Bellman, R., (1957). *Dynamic programming*: Princeton Univ. press.

Bole, B., Tang, L., Goebel, K., & Vachtsevanos, G., (2011). Adaptive load allocation for prognosis-based risk management. *Annual conference of the prognostics and health management society*, pp. 1–10.

Bole, B. M., (2013). *Load allocation for optimal risk management in systems with incipient failure modes*. Doctoral dissertation, Georgia Institute of Technology, Atlanta, USA.

Brown, D. W., Georgoulas, G., Bole, B., Pei, H. L., Orchard, M., Tang, L., Saha, B., Saxena, A., Goebel, K., & Vachtsevanos, G., (2009). Prognostics enhanced reconfigurable control of electro-mechanical actuators. *Annual conference of the prognostics and health management society*.

Brown, D., Bole, B., & Vachtsevanos, G., (2010). A prognostics enhanced reconfigurable control architecture. *Control & Automation (MED), 2010 18th Mediterranean Conference, IEEE*, pp. 1061-1066. doi:10.1109/MED.2010.5547651

Clements, N. S., (2003). *Fault tolerant control of complex dynamical systems*. Doctoral dissertation, Georgia Institute of Technology, Atlanta, USA.

Dorf, R. C., & Bishop, R. H., (1998). *Modern control systems*: Addison-Wesley.

Downes, L., (2015). What's wrong with the FAA's new drone rules. *Harvard Business Review*, Harvard University.

Drozeski, G. R., Saha, B., & Vachtsevanos, G., (2005). A fault detection and reconfigurable control architecture for unmanned aerial vehicles. *Aerospace Conference, IEEE*. doi:10.1109/AERO.2005.1559597

Ge, J., Kacprzynski, G. J., Roemer, M. J., & Vachtsevanos, G., (2004). Automated contingency management design for UAVs. *AIAA 1st Intelligent Systems Technical Conference*, pp. 20–22. doi:10.2514/6.2004-6464

Han, B., & Zhao, G. L., (2004). Course-keeping control of underactuated hovercraft. *Journal of Marine Science and Application*, 3(1), 24-27. doi:10.1007/BF02918642

Hollnagel, E., Woods, D. D., & Leveson, N., (2007). *Resilience engineering: Concepts and precepts*: Ashgate Publishing, Ltd.

Jacobson, D. H., & Mayne, D. Q., (1970). *Differential dynamic programming*. American Elsevier, New York.

Kim, K., Lee, Y., Oh, S., Moroniti, D., Mavris, D., Vachtsevanos, G. J., Papamarkos, N., & Georgoulas, G., (2013). Guidance, navigation, and control of an unmanned hovercraft. *Control & Automation (MED), 2013 21st Mediterranean Conference, IEEE*, pp. 380–387. doi:10.1109/MED.2013.6608750

Orchard, M. E., (2007). *A Particle filtering-based framework for on-line fault diagnosis and failure prognosis*. Doctoral dissertation, Georgia Institute of Technology, Atlanta, USA.

Pannocchia, G., Rawlings, J. B., & Wright, S. J., (2011). Conditions under which suboptimal nonlinear MPC is inherently robust. *Systems & Control Letters*, 60(9), 747-755. doi:10.1016/j.sysconle.2011.05.013

Patron, P., Miguelanez, E., Petillot, Y. R., Lane, D. M., & Salvi, J., (2008). Adaptive mission plan diagnosis and repair for fault recovery in autonomous underwater vehicles. *OCEANS 2008, IEEE*, pp. 1–9. doi:10.1109/OCEANS.2008.5151975

Sconyers, C., Lee, Y., Kim, K., Oh, S., Mavris, D., Oza, N., Mah, R., Martin, R., Raptis, I. A., & Vachtsevanos, G. J., (2013). Diagnosis of fault modes masked by control loops with an application to autonomous hovercraft systems. *International Journal of Prognostics and Health Management*.

Tang, L., Kacprzynski, G. J., Goebel, K., Saxena, A., Saha, B., & Vachtsevanos, G., (2008). Prognostics-enhanced automated contingency management for advanced autonomous systems. *Prognostics and Health Management, International Conference, IEEE*, pp. 1–9, IEEE. doi:10.1109/PHM.2008.4711448

Tang, L., Hettler, E., Zhang, B., & DeCastro, J., (2011). A testbed for real-time autonomous vehicle PHM and contingency management applications. *Annual conference of the prognostics and health management society*, pp. 1–11.

Tran, H. T., (2015). *A complex networks approach to designing resilient system-of-systems*. Doctoral dissertation. Georgia Institute of Technology, Atlanta, USA.

Vachtsevanos, G., Lewis, F. L., Roemer, M., Hess, A., & Wu, B., (2006). *Intelligent fault diagnosis and prognosis for engineering systems*. Hoboken, New Jersey: John Wiley and Sons, Inc.

Yan, Z., Zhao, Y., Chen, T., & Jiang, L., (2012). Fault recovery based mission scheduling of AUV for oceanographic survey. *Intelligent Control and Automation (WCICA), 10th World Congress, IEEE*, pp. 4071–4076. doi:10.1109/WCICA.2012.6359156

Zhang, Y. & Jiang, J., (2008). Bibliographical review on reconfigurable fault-tolerant control systems. *Annual reviews in control*, vol. 32, no. 2, pp. 229–252. doi: 10.1016/j.arcontrol.2008.03.008

BIOGRAPHIES

Sehwan Oh is a Ph.D. candidate in the ASDL at the Georgia Institute of Technology since 2010. He has participated in graduate researches of a turbine engine model regression analysis, Navy transformable ship design, risk analysis of the integration of unmanned aerial vehicle systems into the national airspace system, and smart and sustainable campus design and analysis. His main research areas include resilience system design methodology and control reconfiguration.

Benjamin Lee is an Electrical Engineering graduate student at the Georgia Institute of Technology since 2013. He obtained his Bachelor's degree in Electrical Engineering from the Georgia Institute of Technology in 2013. He has participated in researches of structural health monitoring using acoustic waves, developing and testing of user interfaces for situation awareness in life support systems, and developing tracking control system for a unicycle mobile robot. His main research areas include resilient system design methodology and self-organizational control methods.

Michael Balchanos is research faculty with the School of Aerospace Engineering, where he serves as the Resilient Systems Branch lead at the Aerospace Systems Design Laboratory. His areas of expertise include research work in dynamic systems modeling and simulation methods, as well as SoS-level integration techniques for enabling decision support in complex systems design, involving several applications such as smart energy infrastructures, electric reconfigurable naval ships and unmanned aerial vehicles. He is also leading ASDL's Automotive Systems Research Initiative with applications in electric vehicle energy-based sizing and optimization (EVs) as well as the development of SoS-level frameworks for the connected autonomous mobility ecosystem of the future. He obtained his Diploma in

Physics from the Aristotle University of Thessaloniki, Greece and his M.Sc. and Ph.D. degrees in Aerospace Engineering from Georgia Tech.

Dimitri Mavris is the Boeing Professor of Advanced Aerospace Systems Analysis at the Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, and the director of its Aerospace Systems Design Laboratory (ASDL). Dimitri Mavris received his B.S., M.S. and Ph.D. in Aerospace Engineering from the Georgia Institute of Technology. His primary areas of research interest include: advanced design methods, aircraft conceptual and preliminary design, air-breathing propulsion design, multi-disciplinary analysis, design and optimization, system of systems, and non-deterministic design theory. Dr. Mavris has actively pursued closer ties between the academic and industrial communities in order to foster research opportunities and tailor the aerospace engineering curriculum towards meeting the future needs of the US aerospace industry. Dr. Mavris has also co-authored with his students in excess of 450 publications. Dr. Mavris has received numerous awards and fellowships during his tenure at Georgia Tech. He is currently Fellow of the American Institute of Aeronautics and Astronautics (AIAA) and a Fellow of the National Institute of Aerospace. He was selected as the 1999 Santa Fe Institute Summer fellow. Dr. Mavris is also the recipient of the NSF CAREER award. He has served in several Technical and Program Committees for AIAA and is the US representative to the International Council of the Aeronautical Sciences (ICAS) Board. Academically, Dr. Mavris has been awarded Georgia Tech's prestigious Outstanding Development of Graduate Assistants Award in 1999 and 2004, and in 2000 he received the SAE's Ralph T. Teetor Educator of the Year Award. He is the director of the Center of Excellence in Robust Systems Design and Optimization under the General Electric University Strategic Alliance (GE USA). And he is currently the lead investigator under the Federal Aviation Administration's Center of Excellence under the Partnership for Air Transportation Noise and Emissions Reduction (PARTNER).

George Vachtsevanos is a Professor Emeritus of Electrical and Computer Engineering at the Georgia Institute of Technology. He was awarded a B.E.E. degree from the City College of New York in 1962, a M.E.E. degree from New York University in 1963 and the Ph.D. degree in Electrical Engineering from the City University of New York in 1970. He directs the Intelligent Control Systems laboratory at Georgia Tech where faculty and students are conducting research in intelligent control, neurotechnology and cardiotechnology, fault diagnosis and prognosis of large-scale dynamical systems and control technologies for Unmanned Aerial Vehicles. His work is funded by government agencies and industry. He has published over 300 technical papers and is a senior member of IEEE. Dr. Vachtsevanos was awarded the IEEE Control Systems Magazine Outstanding Paper Award for the years 2002-2003 (with L. Wills and B. Heck). He was also awarded the 2002-2003 Georgia Tech School of Electrical and Computer Engineering Distinguished Professor Award and the 2003-2004 Georgia Institute of Technology Outstanding Interdisciplinary Activities Award.