# Scope and Application of Reconfigurable Control

Vedran Bakarić, Zoran Vukić, *Member, IEEE*, and Radovan Antonić, *Member, IEEE*

***Abstract*****-- Reconfigurable control is an emerging area of control design, tightly connected with adaptive, fault-tolerant and intelligent control. The paper defines the goals of reconfigurable control system design, describes structure of complex and simplified reconfigurable control systems, introduces fault-tolerant control, and presents two interesting reconfigurable control approaches: neural direct adaptive control design and control allocation reconfiguring problem. In the conclusion, some thoughts about application to unmanned underwater vehicles are given.**

***Index Terms*****-- Adaptive systems, Fault-tolerant systems, Flight control, Reconfigurable control, Unmanned underwater vehicles**

## I. Introduction

The task of reconfigurable control is to achieve an automated, quick control system reconfiguration as a reaction to a sudden or large change which appeared in the controlled system or its surrounding. The change which requires such intervention may be of unexpected and unwanted kind, as a failure, but planned and commanded interventions related to variable operating conditions or restating of control goals and specifications can be also considered. The reconfiguration of the control system takes the form of the structural or parametrical change inside the control system. It may consist of greater or smaller changes in various parts of the control law, switching on or off various software modules inside the control system, changes in configuration of sensors and actuators being used, or re-specification of high-level control objectives. The ultimate goal of the control system reconfiguration is to gain as much of control performance as possible after the change that occurred in the controlled system. At least, critical performance indices related to system stability and safety should be maintained [1], [2].

Reconfigurable control is needed primarily for the systems which should avoid emergency shutdown due to faults as long as possible, and for the systems which should smoothly and automatically cycle through many vastly different operating modes and even physical configurations during their operation. Most of research activity is currently associated with advanced aircraft control. Besides aircrafts, important areas of application are other advanced vehicles, robotic systems and manipulators, and industrial process control.

V. Bakarić is with Brodarski Institute, Av. V. Holjevca 20, 10020 Zagreb, Croatia (e-mail: bakaric@hrbi.hr)
Z. Vukić is with Department of Control and Computer Engineering in Automation, Faculty of Electrical Engineering and Computing, Unska 3, 10000 Zagreb, Croatia (e-mail: zoran.vukic@fer.hr)
R. Antonić is with Split College of Maritime Studies, Zrinsko-Frankopanska 38, 21000 Split, Croatia (e-mail: antonic@pfst.hr)

## II. Basic Reconfigurable Control Schemes

In fully developed complex control system architecture reconfigurable control lies in the middle level of the control hierarchy (Fig. 1) [1]. The lower level contains ordinary control mechanism, which is to be reconfigured, while the upper level monitors overall progress of the controlled system and decides about its further tasks in accordance with current conditions and the prescribed plan. Complex hierarchical control structure of this sort is often called *intelligent control architecture* and, besides reconfigurable control and associated functions, it comprises many other modules that are not shown in Fig. 1.

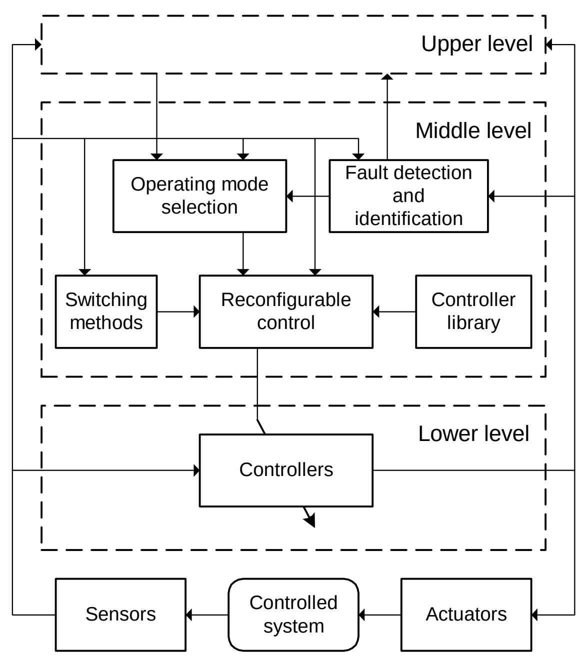


Fig. 1. Reconfigurable control as a part of a complex intelligent control system architecture.

The need for controller reconfiguration should be recognized by monitoring of the controlled system, its environment and mission progress. Reconfiguring action can be *deliberative* – result of a high-level command from either the upper level of the control system or human operators, or *reflexive* – a quick automated default reaction to some observed change. In either case, the requests for reconfiguring intervention arrive from the upper hierarchical level or some block for monitoring and analysis (e.g. fault detection and identification module). The operating mode selector receives such requests, makes ultimate decision to accept them, supplies details about exact type, moment and method of reconfiguring action if not yet specified, and finally starts the reconfiguring process. Reconfiguring block drives the reconfiguring action, using information from the operating mode selector, sensory output of the controlled system, some auxiliary functions as the switching manager with transient management system, libraries of prepared controller structures and parameters, and on-line controller design and parameter tuning algorithms. Besides controllers, other modules of the intelligent control architecture – sensors, actuators, data processing filters, fault detection methods, mission management processes etc. – may be changed and switched as well.

Research of such fully-developed reconfiguration architecture is not an easy task. As mentioned, reconfigurable control is part of hierarchical intelligent control architecture, which is in reality a distributed and modular system, comprising many different multi-task processing units and software modules. State-of-the-art control platform with associated programming tools is needed for advanced research and further practical design of such complex control systems, including their reconfigurable control [1]. Core development with such platforms is centered about compatibility, interface standardization and communication methods between the modules, essential to enable smooth and reliable connecting and re-connecting of numerous hardware and software components, which is needed for both initial system integration and later reconfiguring actions.

Reconfigurable control system takes much simpler form if only one specific reconfiguring action, rather than generic framework for all possible reconfigurations, is considered [2], [3]. Such research usually concerns control reconfiguring action as a fault accommodation procedure within fault tolerant control system (Fig. 2). Reconfiguring action is here purely reflexive reaction to some observed faulty conditions, not part of a prepared mission plan. Reconfiguring block is hierarchically subordinate to the fault identification logic, which takes the role of the operating mode selector.

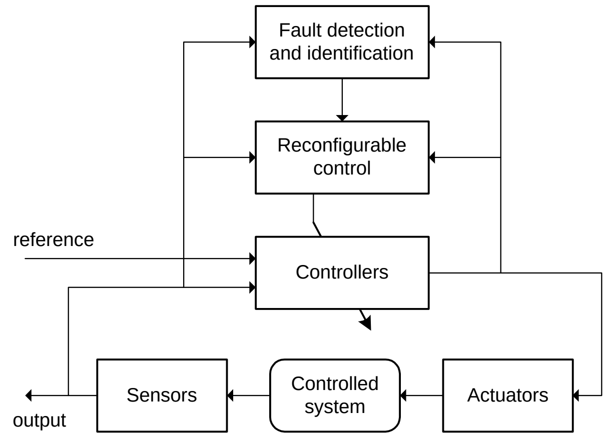


Fig. 2. Reconfigurable control as a part of a fault tolerant control system.

Depending on the type of the anticipated faults and other changes, reconfiguring intervention can be aimed at guidance, control, sensors or actuators, i.e. reference generating algorithms, control algorithms, sensory data processing algorithms (data fusion filters), or actuator control allocation algorithms (mixers). Redundancy – hardware redundancy and/or analytical redundancy via appropriate mathematical models – is essential for changing sensor configuration. Faulty or currently not useful sensors may be simply switched off, while more elaborate methods include weighted fusion of sensory signals to take into account variations and differences in reliability and noise level [4]. Similarly, reconfigurable actuator structures are redundant – the number of actuators is greater than the number of controlled variables, and the system remains controllable if any single actuator becomes inoperable [6], [5]. Appropriate actuator configuration is found by some optimization method.

Reconfiguring action takes the form of parametric change, switching inside a bank of different modules of the same type, or switching with subsequent parametric adaptation. Module switching methods accomplish quick reaction to the perceived change and avoid more complicated on-line design algorithms, but they need comprehensive design research (to prepare appropriate modules and their parameters for anticipated operating modes) and substantial computer memory (to store module libraries). Common parameter adaptation methods are linear, using linear controllers and system models. Greater potential scope of application is possible with non-linear parametric adaptation or signal correction, provided by neural networks or some other approach. While module switching operation is a discrete event-driven action that should not be performed too often, module parameters may continuously change in response to perceived variable controlled system condition.

Quite often simple reconfigurable fault-tolerant control methods go one step further from the architecture shown in Fig. 2 and integrate the fault detection and identification block and the reconfiguring block in such way that there is no more explicit fault identification step, but only some condition monitoring and adaptive reaction to it. Such reconfiguring method can be included in the intelligent control architecture (Fig 1) as well, but now it would reside at the lower hierarchical level and be considered as a part of an advanced ordinary controller. Fully-developed reconfigurable control architecture can be then added at the middle level as in Fig. 1, and perform switching of advanced low-level controllers and other modules. Direct neural adaptive control, described in section IV.A of this paper, is a good example of such approach [6], [7].

## III. Fault Detection and Identification

Fault tolerant control system performs fault detection, isolation, identification and accommodation. *Fault detection* (or *condition monitoring*) monitors the controlled system and detects if there is any failure, *fault isolation* finds the location of the failure, *fault identification* supplies quantitative parameters of the failure, and *fault accommodation* provides control system adaptation to the failure. Fault accommodation can be either *passive* (control system is designed to be robust to anticipated faults) or *active* (control system is being reconfigured after the fault is detected). *Redundancy*, physical or analytical, is in the core of the fault tolerant control since the fault cannot be by-passed if there is no alternative signal propagation channel [2], [3].

Basic fault detection technique relies on simple *signal monitoring*. Checking of range (minimum and maximum) and rate (speed of change) of a measured signal generates clear fault symptoms when the signal takes forbidden or even impossible values or dynamics. Redundant sensors and processing units may be checked against each other. Special additional sensors can be used to monitor actuators and other critical subsystems to check if they behave correctly.

More advanced is *model-based fault detection*. Here the fault detection block compares the monitored system with its analytical model (so-called *analytic redundancy* - Fig. 3). *Residuals* are formed from differences between the measured behavior of the monitored system and the calculated or tabulated proper behavior provided by the analytical model. A fault symptom would be generated when a residual exceeds its prescribed threshold. Residuals should be sensitive to faults to provide *fault detectability*, but also *robust* (not over-sensitive to disturbances, measurement noise and modeling errors) and able to provide *fault isolability* (different residual patterns should be generated for different fault modes). Subsequent step of fault isolation and identification uses fault symptoms to locate the detected fault, i.e. activate the appropriate *fault mode*, and quantify it. It can be rather simple process if advanced residual generators that prepare unique fault symptoms from complex residuals are used.

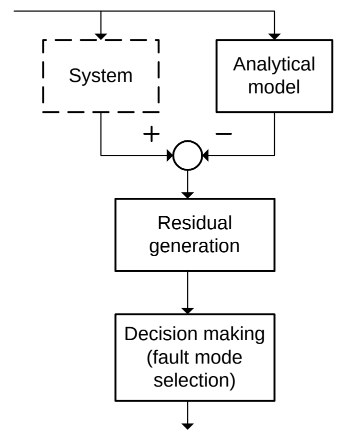


Fig. 3. Residual generation principle.

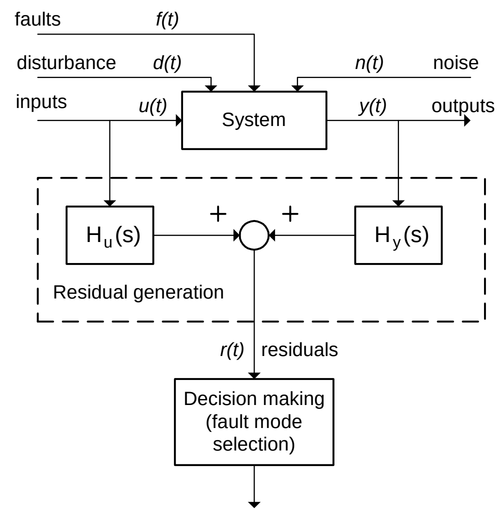


Fig. 4. Block diagram of general linear residual generator.

Generated residual can be based upon measured and estimated system state (*observer-based residual generators*), identified and prescribed parameters of the analytical model, or redundancy relations of the system formed by the analytical model (*parity space approach*). However, residual generator may be considered simply as an algorithm that should generate good residuals using inputs, outputs and analytical model of the monitored system as its input information. With this in mind, general linear residual generating scheme is given in Fig. 4.

Classification methods, neural networks, fuzzy logic and expert systems with heuristic knowledge form core of *model-free methods* for fault detection. In some cases it can be said that a *qualitative model* of the monitored system, based on heuristic observations, is used instead of an analytic one, based on mathematical identification. Contrary to model-based fault detection, fault symptoms can here be rather simple observations, which are subsequently processed in advanced fault identification mechanism.

Model-free methods are not restricted by condition of having accurate analytical model of the monitored system. Moreover, they avoid residual robustness problems of model-based fault detection. On the other hand, model-based fault detection is more mature, and it generally needs by far smaller amounts of computing power and memory.

## IV. Some Examples of Reconfigurable Control

### A. Direct Neural Adaptive Control

Direct adaptive control architecture using neural networks is recognized as a very promising method for aircraft flight control [6]. The architecture is based on feedback linearization principle, which has been also successfully applied for other dynamic systems besides aircrafts, including ships, underwater vehicles and robot manipulators. The dynamic system to be controlled is of the form

$$\ddot{\mathbf{x}} = f(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}) \tag{1}$$

where $\mathbf{x}$ is the state vector, $\mathbf{u}$ the control vector and $f(\cdot)$ the non-linear function representing the system model. Dimensions of the control and state vectors $\mathbf{x}$ and $\mathbf{u}$ must be identical (square system). If the pseudo-control variable $\mathbf{v}$ is introduced as

$$\mathbf{v} = \ddot{\mathbf{x}} \tag{2}$$

then the control vector $\mathbf{u}$ can be calculated as

$$\mathbf{u} = \hat{f}_{inv}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{v}_{\mathrm{S}}) \tag{3}$$

where $\mathbf{v}_{\mathrm{S}}$ is some suitable expression selected for the pseudo-control $\mathbf{v}$, and $\hat{f}_{inv}(\cdot)$ some available estimation of the non-linear function $f_{inv}(\cdot)$, which is the inverse of the system model $f(\cdot)$. In the considered case [6], the pseudo-control $\mathbf{v}_{\mathrm{S}}$ is assembled from three terms: feedforward acceleration tracking $\ddot{\mathbf{x}}_{\mathrm{d}}$, linear proportional and derivative (PD) state feedback $\mathbf{v}_{\mathrm{PD}}$, and inversion error correction term $\mathbf{v}_{\mathrm{ad}}(\cdot)$

$$\begin{aligned} \mathbf{v}_{\mathrm{S}} &= \ddot{\mathbf{x}}_{\mathrm{d}} + \mathbf{v}_{\mathrm{PD}} - \mathbf{v}_{\mathrm{ad}}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u}) \\ \mathbf{v}_{\mathrm{PD}} &= \mathbf{K}_{\mathrm{D}}(\dot{\mathbf{x}}_{\mathrm{d}} - \dot{\mathbf{x}}) + \mathbf{K}_{\mathrm{P}}(\mathbf{x}_{\mathrm{d}} - \mathbf{x}) \end{aligned} \tag{4}$$

The desired state vector $\mathbf{x}_{\mathrm{d}}$ and its derivatives $\dot{\mathbf{x}}_{\mathrm{d}}$ and $\ddot{\mathbf{x}}_{\mathrm{d}}$ are formed by closed-loop response models from control reference signals. PD control parameters $\mathbf{K}_{\mathrm{P}}$ and $\mathbf{K}_{\mathrm{D}}$ are selected to minimize the control errors, assuming that the inversion errors are fully compensated. The inverse system model $\hat{f}_{inv}(\cdot)$ is implemented as a static (non-adaptive) neural network. And finally, the inversion error corrections $\mathbf{v}_{\mathrm{ad}}(\cdot)$ are provided by an adaptive neural network. The algorithms for network adaptation with stability proof are presented in [6].

For the flight control application, the control structure based on the method described above is decoupled into three single-input single-output control channels: roll, pitch and yaw. The yaw and pitch channels are modeled by second-order differential equations, while the roll channel takes the simpler form of the first order differential equation, as follows

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, u_1) \\ \ddot{x}_i &= f_i(\dot{x}_i, x_i, u_i) \quad , \quad i = 2,3 \end{aligned} \tag{5}$$

As reference inputs, the control channels accept acceleration or velocity commands $\dot{x}_{cd1}$, $\ddot{x}_{cd2}$ and $\ddot{x}_{cd3}$ provided by the aircraft pilot. The control structure of an individual channel is described by (4) and (3), as shown on Fig. 5. The reference response model takes the form of a simple linear second-order or first-order (for the roll channel) filter. Each channel uses its PD (or P, for roll channel) time-invariant feedback controller, static neural inverse model and adaptive neural inversion correction element. Linear-in the-parameters sigma-pi network and multilayer perceptron has been tested for the adaptive inversion correction problem: it was found that more complex multilayer perceptron gives better results and so it is worth the increased computing effort. The control allocation manager, or mixer, distributes demanded control activity of the three control channels $u_i$ to $s$ control effectors (fins, rudders and ailerons) $\delta_j$ ( $s > 3$ ).

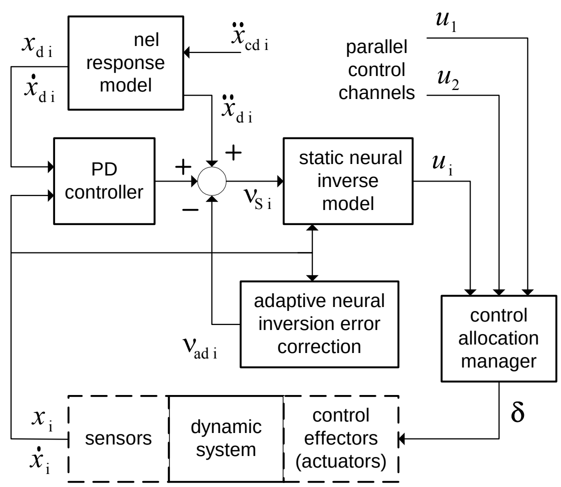


Fig 5. Neural feedback linearization control structure for the $i$th control channel of the flight control system.

Neural adaptive control based on feedback linearization is a very powerful adaptive control method. The adaptation of inversion error correction is quick enough to adequately response to severe sudden faults, hence this method is also introduced as a fault-tolerant reconfigurable control. Compared with other methods, it does not need separate fault detection and fault isolation steps, nor quick system re-identification for the controller re-design, nor excessive look-up tables with prescribed control structures for different fault modes. However, feedback linearization is applicable only to certain classes of systems. The demand for equal dimensions for the state and the control vector $\mathbf{x}$ and $\mathbf{u}$ in (3) is not critical since it can be often circumvented by appropriate control signal transformations within the control allocation manager (Fig 5). Much more difficult are demand for the existence of a reasonably good inversion model $\hat{f}_{inv}(\cdot)$ in (3), and determination of the state and reference derivatives $\mathbf{x}$ and $\mathbf{x}_{\mathrm{d}}$ in (4).

### *B. Actuator Reconfiguration Methods*

Reconfigurable actuator sets are redundant, i.e. the number of active actuators is greater than the number of controlled variables. Quite often they are handled by control allocation managers (mixers, matrices) [3], [5]. Such managers distribute control signals $\mathbf{u} \in \Re^m$ provided by the control system to the actual actuator signals $\boldsymbol{\delta} \in \Re^s$ , $s > m$ (Fig. 6). In the case of the linear control problem, the control allocation manager is the matrix $\mathbf{K}$ that transforms $\mathbf{u}$ into $\boldsymbol{\delta}$

$$\boldsymbol{\delta} = \mathbf{K}\,\mathbf{u} \tag{6}$$

while the controlled system is modeled as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\boldsymbol{\delta} = \mathbf{A}\,\mathbf{x} + \mathbf{B}_{\mathrm{C}}\,\mathbf{u} \tag{7}$$

where $\mathbf{B}_{\mathrm{C}} = \mathbf{B}\,\mathbf{K}$ .

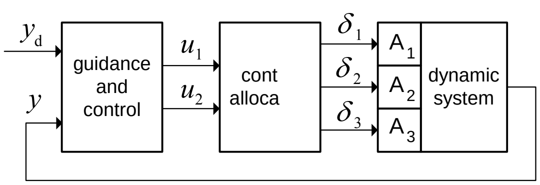


Fig. 6. Control allocation.

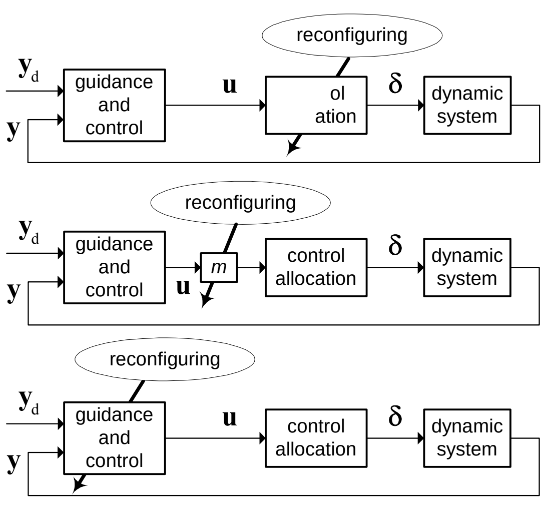


Fig 7. Possible places of intervention in the control architecture with the aim of actuator reconfiguration.

The most natural way to make actuator reconfiguration is to change the control allocation law, but it is also possible to modify the existing control signals, or even to change the control law itself (Fig 7). Control allocation design is basically an optimization problem since the problem solution is not unique. Consequently, actuator set reconfiguration can be achieved as a re-optimization of control allocation, taking into account the changes (actuator failures and other operating mode changes) that prompted reconfiguration. However, the involved calculation may be too complex for the on-line design, particularly in the case of advanced optimization methods and time-critical fault accommodation scenarios. Thus, off-line optimization is often invoked, and its results prepared in the form of look-up tables. Besides quick reconfiguration, off-line design methods enable extensive validation and verification needed for safety critical applications, as aircraft flight control. On the other side, they need an excellent system model for off-line research and large amounts of computer memory for look-up tables.

Simplified actuator reconfiguration problem statement uses the nominal actuator system behavior as its optimization goal. This is the pseudo-inverse approach, which in its simplest form goes as follows. Consider the system model given by (7). Due to a fault in the actuator system, the control matrix $\mathbf{B}_{\mathrm{C}}$ has changed from nominal $\mathbf{B}_{\mathrm{Cn}}$ to some faulty $\mathbf{B}_{\mathrm{Cf}}$ . The new control signal $\mathbf{u}_{\mathrm{f}}$ is calculated from nominal control $\mathbf{u}_{\mathrm{n}}$ to minimize the difference between the nominal control action $\mathbf{B}_{\mathrm{Cn}}\mathbf{u}_{\mathrm{n}}$ and the new one $\mathbf{B}_{\mathrm{Cf}}\mathbf{u}_{\mathrm{f}}$ . The optimized solution is

$$\mathbf{u}_{\mathrm{f}} = \mathbf{B}_{\mathrm{Cf}}^{\dagger}\,\mathbf{B}_{\mathrm{Cn}}\,\mathbf{u}_{\mathrm{n}} \tag{8}$$

where

$$\mathbf{B}_{\mathrm{Cf}}^{\dagger} = (\mathbf{B}_{\mathrm{Cf}}^{\mathrm{T}}\,\mathbf{B}_{\mathrm{Cf}})^{-1}\,\mathbf{B}_{\mathrm{Cf}}^{\mathrm{T}} \tag{9}$$

is the pseudo-inverse of the matrix $\mathbf{B}_{\mathrm{Cf}}$ .

Actuator failure implies partial or total loss of an actuator, but it can have even more adverse effect if the failed actuator continues to improperly actuate the controlled system. Such is the case with a stuck, or jammed, control surface of the flight control system. Actuator reconfiguration scheme now has not only to re-allocate control action to the remaining actuators, but also to compensate disturbance caused by the failed actuators. Pseudo-inverse approach can be rather easily extended to address this problem as follows. Consider the controlled system given by (7). After the actuator failure, the system can be modeled as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}_{\mathrm{f}1}\,\boldsymbol{\delta}_{\mathrm{f}} + \mathbf{B}_{\mathrm{f}2}\overline{\boldsymbol{\delta}} \tag{10}$$

where the control matrix $\mathbf{B}$ is divided into $\mathbf{B}_{\mathrm{f}1}$ connected with usable actuators and $\mathbf{B}_{\mathrm{f}2}$ connected with failed actuators; $\overline{\boldsymbol{\delta}}$ represents uncontrollable actuator input due to failure, and $\boldsymbol{\delta}_{\mathrm{f}}$ new actuator control signal. Pseudo-inverse is achieved when (7) and (10) are equalized. Optimal selection of $\boldsymbol{\delta}_{\mathrm{f}}$ is

$$\boldsymbol{\delta}_{\mathbf{f}} = \mathbf{K}_{\mathrm{f}}\,\mathbf{u} + \mathbf{d}_{\mathrm{f}} \tag{11}$$

with

$$\mathbf{K}_{\mathrm{f}} = \mathbf{B}_{\mathrm{f}1}^{\dagger}\,\mathbf{B}\,\mathbf{K} \quad , \quad \mathbf{d}_{\mathrm{f}} = -\mathbf{B}_{\mathrm{f}1}^{\dagger}\,\mathbf{B}_{\mathrm{f}2}\,\overline{\boldsymbol{\delta}} \tag{12}$$

Both pseudo-inverse approach and more complex optimization methods need precise information about the actuator system change in order to reconfigure actuator set. Fault identification step is thus here unavoidable in fault tolerant control applications.

### *C. Underwater Vehicles*

From dynamic control standpoint, two basic types of unmanned underwater vehicles (UUVs) are cruising and hovering vehicles [8], [9]. Cruising vehicles are used for oceanic data acquisition, survey and search, and they operate while continuously moving forward. For motion control they besides main propulsion thruster usually use control surfaces, which are more efficient than maneuvering thrusters at significant forward speeds, but unusable when the vehicle is not in forward motion. The control problem is often decoupled into several control channels: forward speed, steering, diving and rolling. Meanwhile, hovering vehicles perform tasks of underwater inspection, intervention (manipulative work) and intervention monitoring, and so they must be able to hold station and move backwards or sideways, but they do not need the forward speed range desired for cruising vehicles. Consequently, they maneuver using maneuvering thrusters, have more degrees of motion freedom and demand greater control accuracy than cruising vehicles. Decoupled control design is here possible only for rather simple cases, but some hydrodynamic effects pronounced at high speeds may be often neglected.

Reconfigurable control examples considered in sections A and B of this chapter are concerned with aircraft flight control applications. Although typical representative vehicle state vectors are not identical, decoupled cruising UUV control has many similarities with an aircraft flight control system – the main difference is much greater influence of hydrodynamic and hydrostatic forces and moments. However, the number of control effectors at an aircraft is usually by far greater than at a small cruising UUV, while hovering UUVs have more actuator redundancy. Furthermore, reconfiguration speed after a failure is not so critically important. On the other hand, untethered UUVs have to economize with maneuvering power more than almost any aircraft, since it is supplied to them from on-board batteries or fuel cells. And finally, UUVs and unmanned aerial vehicles have different control architecture than piloted aircrafts.

With all this in mind, it can be seen that the demands for reconfigurable control are somewhat different between aircrafts and UUVs. While the basic idea – automatic fault accommodation and adaptive operating mode change – remains the same, important application priorities are different. For aircraft, the most important is quick and safe automatic fault accommodation; for UUV, operation with partial failures and optimization of vehicle performance through different operating modes have more significance. Direct neural adaptive control and similar methods are interesting for cruising UUV application. They handle both external disturbance adaptation and fault accommodation without extensive vehicle modeling, and were successfully used for rather similar control problems. Control allocation reconfiguration is interesting primarily for systems with ample actuator redundancy. In UUV control area they include most hovering and some cruising vehicles. However, maybe the most advanced reconfigurable control applications for UUVs are methods for sensor set reconfiguring, very useful for both fault tolerant control and operating mode switching [4].

## V. CONCLUSION

The role of reconfigurable control will be substantially increased as more complex control systems are introduced and ever greater degree of performance, safety and autonomy desired. The main research areas and some interesting control reconfiguring methods are presented in this paper. Some other methods and hot research areas are also worth of mentioning, including geometrically reconfigurable systems (robots and robotic manipulators), management and suppression of switching transients, and reconfigurable manufacturing systems.

## REFERENCES


[1] L. Wills, S. Kannan, S. Sander, M. Guler, B. Heck, J. V. R. Prasad, D. Schrage, and G. Vachtsevanos , "An open platform for reconfigurble control," *IEEE Control Syst. Mag.*, vol. 21, no. 6, pp. 49-64, June 2001

[2] Q. Zhao, "Fault tolerant control systems," university talk, Univ. of Alberta, 2000, Available: http://www.ee.ualberta.ca/~qingzhao/u_talk/index.htm

[3] Y. Huo, P. A. Ioannou, and M. Mirmirani , "Fault-tolerant control and reconfiguration for high performance aircraft: rewiev," CATT, Univ. of Southern California , Los Angeles, Tech. Rep. No. 01-11-01, 2002 (Available: http://www.usc.edu/dept/ee/catt/2002/ying/web_AIR/ )

[4] L. Drolet, F. Michaud, and J. Côté ,"Adaptable sensor fusion using multiple Kalman filters," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Takamatsu, Japan, 2000

[5] M. C. Cotting and J. J. Burken, "Reconfigurable control design for the full X-33 flight envelope," NASA Dryden Flight Res. Center, Edwards, CA, Rep. No. H-2458, 2001 (also in *AIAA Guidance, Navigation & Control Conf.*, Montreal , Quebec, Aug. 2001.)

[6] A. J. Calise, S. Lee, and M. Sharma , "Development of a reconfigurable flight control law for the X-36 tailless fighter aircraft," in *AIAA Guidance, Navigation and Control Conf.*, Denver, CO, Aug. 2000

[7] S. Kannan, C. Restrepo, I. Yavrucuk, L. Wills, D. Schrage, and J. V. R. Prasad , "Control algorithm and flight simulation integration using the Open Control Platform for unmanned aerial vehicles," in *AIAA Digital Avionics Systems Conf.*, St. Louis, MO, Oct. 1999.

[8] K. P. Valavanis, D. Gracanin, M. Matijasevic, R. Kolluru, and G. A. Demetriou, "Control architectures for autonomous underwater vehicles," *IEEE Control Syst. Mag.*, vol. 17, no. 12, pp 48-64, Dec. 1997

[9] T. I. Fossen, *Guidance and Control of Ocean Vehicles*, Chichester, UK: J. Wiley & Sons, 1994.