Functional degradation of cyber-physical aerial system for trajectory planning and agent tracking

D Sayfeddine

1 PhD, South Russian State Polytechnic University, 132, Prosveshenie Str., Novocherkassk, Russia
E-mail: daher@live.ru

Abstract. With the development of cyber physical systems, new methods were initiated to overcome failures in both cyber and physical domains. Failures can be numerous since the cyber side faces real-time execution problems and bugs in the algorithms. On the other hand, the physical parts such as actuators and sensors are in direct contact with the surroundings and their performance strictly depends on changing in the circumstances. Developing methods that can overcome a combined cyber-physical failure are of great importance for the survivability of autonomous systems. This paper addresses the functional degradation of UAV as a cyber physical system during the tracking process of a mobile agent using possible combinations of AI, optical odometry and range finding techincs.

1. Introduction
A cyber physical system is a result of advances in the artificial intelligence. It consists of reliable fusion between the computational and physical platform embedded within the same machine; however, it enables better interaction capabilities with the surrounding (world and a human operator) through many modalities and processes. In light of that, the terms of interaction, integration and enhancement are key areas where the research of cyber physical systems is occurring and developing [7].

In relation to autonomous robotic concepts, the cyber physical system is likely to have an impact on future design, thus achieving new functionalities, better safety performance and opportunity for wider integration.

Pertaining to unmanned aerial vehicles, cyber physical system can be a game changer. Since the number of aerial platforms is increasing, the need to establish reliable platforms governing the design, integration and exploitation of such system has becomes a priority. As a result, one of the key technical challenges is the verification and validation of autopilot systems with special consideration of flight’s sustainability and the safety of possibly impacted bodies (surrounding UAVs, airplanes, creature, etc. [1,6])

In this regard, the scale of the data to be validated will exponentially increase. As a result, the cost of design becomes unjustifiable in order to have global implementation. Endorsement of this statement can serve the cost of 5th generation stealth fighters, where the number of airplane released or forecasted does not justify the performance margin and the cost embedded in the design.

On the other hand, researching cyber physical systems for miniature aerial vehicle is more cost and time effective. The final aim can be limited to optimizing path-planning, validation of autopilot critical systems taking into consideration the safety factor requirement starting from the design stage throughout the implementation life cycle. As a result, trade-offs between system complexity and validation
simplicity have to be established. Hence, the following challenges to be considered during flight control are highlighted below:

- autonomous piloting with degradation modes;
- real time diagnostic of the actuating parts (possibility to shift between actuators to maintain safe flight regime)
- data fusion with reference to IoT (Internet of Things)

2. Case study

Performance is a key parameter when evaluating the success of a cyber physical system. When it is inevitable to substantially run the system, a priority must be set to cut down on less important tasks and focus on sustainability and completion of the aim of such system. The compromise on some processes is called graceful or functional degradation.

There is a variety of reasons behind the drop of performance of the cyber-physical system – it can be mechanical, bugs in the algorithms or even external conditions.

Concerning to the UAV topic, mechanical failures are catastrophic. On the other hand, programming failures such as escaped mistakes might lead to undesirable results. External conditions play decisive role in making both the mechanical and algorithmic parts function as intended to be. The paper will focus on this topic and address possible mitigation plan as an option for graceful degradation.

The case study is built around the following scenario – a GPS guided UAV tracking the mobile agent (a car with random trajectory) is facing a connection problem for certain reasons. The task changes from “tracking while scanning” to “identifying while scanning”. The difference between the two tasks is basically the sequence of procedures to be taken into account in order to track the moving agent. In the first case, the position of the agent is tracked using GPS. The coordinates are sent to the autopilot in order to correct the trajectory of the UAV using feedback signal from the laser rangefinder. In the second case, the UAV has to identify the agent using optical odometry and track the generated trajectory. Time-wise, the first case is a fast process and does not impose overloading on the built-in memory resources.

By reviewing the literature, we can say the topic has been studied from several angles. Indeed, researches about reliability of trajectory planning methods, coverage tasks, velocity and stability optimization technics are widely available, however, the same was presented as a deliverable for isolated micro engineering specialties. The need to embed all the aforementioned studies in a single validation case is important. This will change the status of proposed automation solution from “being theoretically founded” to be “actually realized”.

Firstly, let’s identify the system model using sets of mapping constraints. Each application is modelled using a graph function $G_T$ representing combinations between the available hardware and the tasks $T_T$ that can be achieved with. It is obvious that that a single hardware cannot achieve the desired output, hence, logic connections between different hardware should be taken into consideration. The same can be denoted as boundary $B_T$. Similarly, a graph illustrating the active architecture $G_A$ is represented by the active resource $R_A$ (such as memory, active bus, links etc.) and the boundary between them $B_A$.

In light of the above, a functional degradation task is set to find the optimal combination achieving a given goal using space exploration, defining a common boundary $B_{SE}$ between $B_T$ and $B_A$. Hereafter, an architectural resource is selected and a task $T_T$ is identified.

Figure 1 represents the explained model [2,7,8].
In the illustration above, the indexes represent the following notions: i and j – task types, the dotted line is a possible connection serving as an actual physical connection, and the arrow orientation represents a sequence of finish-to-start scheme.

By applying the same principle on the case study, in case of GPS failure, the UAV still have resources such as the camera and the range finder. The link between the two hardware is established internally through the cyber-physical system. The resources available are the on-board processing memory and the communication link between all hardware. The task that can be done separately i.e. the function of each device is measuring range between the UAV and the mobile agent, snapping real-time video frames using the camera and processing the data using the on-board software.

According to the notions of the functional degradation, a solution is viable only if each task at least is launched once and data are flowing consistently. In addition, to run the degraded system online, the computation memory should not exceed the available limits on boards. Additional constraints should also be considered such as the unpredictability of the tracked agent trajectory, the effect of the variation in the luminosity level during the identification process of the agent and the efficiency of the range finder.

3. Optical odometry
The concept of optical odometry is founded on calculating the distance between an object and a tracking hardware using trigonometric approaches and frames calculations. According to the case study scenario, the camera will be used to track a mobile agent taking into consideration the variation of luminosity. The result of the identification process will serve to correct the readings of the range finder. However, in case of detection of identical agents at the same frame, the task becomes difficult to handle. The solution will consist in transforming the task to extrapolation of the movement of the required agent. This transformation satisfies the aforementioned notions of graceful degradation in terms of periodic running of all components and the need to have a permanent flow of data between different hardware. It also minimizes the consumption of computation power required for agent tracking.

The role of the extrapolation is to interpret the historical data and to propose a possible subsequent movement of the agent without having to identify it for all the captured frames. This will be important factor to consider as the processing time of the identification and tracking can exceed the frequency of generated frames per second.

3.1. Extrapolation architecture
The extrapolation model is based on artificial neural network with exogenous input (NARX). Figure 2 depicts the functionality of the proposed method. The position of the agent and its movement is represented with the red curve (time series). The historical positions are retrieved from the curve and acts as input perceptrons (red circles). An additional input perceptron represents the data acquired via the range finder and the camera with a frame shifting. The data are analyzed and an output perceptron generates the extrapolated position of the mobile agent.
Figure 2. Extrapolation methodology using NARX

The nonlinear autoregressive NARX is a model of nonlinear neural network, which can accept dynamic inputs represented via time-series sets. This is the main advantage of the NARX over feed-forward back-propagation neural networks [9, 10]. NARX can deal also with discrete and continuous inputs [11].

The training of the NARX is achieved using BPTT because the output of the network is not fed into the tracking algorithm. Hence, the training is done passively following three steps:

1- The first step consists of calculating and discovering the status of the activation functions \( x(n) \) of each perceptron starting from \( u(n), x(n-1) \) and \( y(n-1) \) or the activation of the output layer if it is fed into a certain perceptron.

2- The second step includes the calculation of the backpropagation error of each perceptron starting from \( n = T \), \( x(n) \) and \( y(n) \) for each instance of time \( n \). This is achieved using the following system of equations (2):

\[
\begin{align*}
\delta_j(T) &= (d_j(T) - y_j(T)) \frac{\partial f(u)}{\partial u} \bigg|_{u=z_j(T)} \\
\delta_i(T) &= \left[ \sum_{j=1}^{L} \delta_j(T) w_{ji}^{out} \right] \frac{\partial f(u)}{\partial u} \bigg|_{u=z_i(n)} \\
\delta_j(n) &= (d_j(n) - y_j(n)) \sum_{i=1}^{N} \delta_i(n+1) w_{ji}^{back} \frac{\partial f(u)}{\partial u} \bigg|_{u=z_j(n)} \\
\delta_i(n) &= \left[ \sum_{j=1}^{N} \delta_j(n+1) w_{ji}^{out} \right] \frac{\partial f(u)}{\partial u} \bigg|_{u=z_i(n)}
\end{align*}
\]

where \( \delta_j(T) \) — the backpropagation error of the output perceptron, \( \delta_i(T) \) — the backpropagation error of the perceptron located in the hidden layer with activation \( x_i(T) \), \( \delta_j(n) \) and \( \delta_i(n) \) are consequently the backpropagation error of the output perceptron and the one located in the hidden layer in an earlier time \( T \) layer and \( z_i(n) \) — the potential of each perceptron.

\[
\begin{align*}
w_{ij} &= w_{ij} + \gamma \sum_{n=1}^{T} \delta_i(n) x_j(n-1) \\
w_{ijn} &= w_{ijn} + \gamma \sum_{n=1}^{T} \delta_i(n) u_j(n)
\end{align*}
\]
3- After finding the backpropagation error, the weights connecting different perceptron are calculated using the following system:

\[
\begin{align*}
    w_{ij}^{\text{out}} &= w_{ij}^{\text{out}} \gamma \left( \sum_{n=1}^{T} \delta_i(n) u_j(n) \text{ if } j - \text{ an output perceptron} \right) \\
    w_{ij}^{\text{back}} &= w_{ij}^{\text{back}} + + \gamma \sum_{n=1}^{T} \delta_i(n) x_j(n) \text{ if } j \text{ is a hidden perceptron} \\
    w_{ij}^{\text{back}} &= w_{ij}^{\text{back}} + + \gamma \sum_{n=1}^{T} \delta_i(n) y_j(n - 1)
\end{align*}
\]

where \( w_{ij} \) is the weight connecting the hidden perceptron; \( w_{ij}^{\text{in}}, w_{ij}^{\text{out}} \) and \( w_{ij}^{\text{back}} \) are the input, output and feedback weights consequently; \( \gamma \) is an incremental small value that is used during the minimization of the squared error.

3.2. Degradation-aware architecture

To identify whether the degradation is occurring in lieu of a substantially functional system, a diagnostic architecture has to be established. The diagnostic result is a status function \( f(\rho, \tau) \) where \( \rho \) represents Boolean status of the selected resource \( R_A \) and \( \tau \) is the activation of a selected task \( T_T \). All possible combinations of the tasks and resources are coded using a binary vector for which every component has a Boolean value. Equation below describes the status function in relation with the binary vector:

\[
f(\rho, \tau, V) = \bigwedge_{v \in V} \left[\neg v \bigvee_{v \in V} f_v(\rho, \tau) \right] \bigwedge_{r \in \rho} \delta_r(\tau)
\]

where \( V \) is the binary vector; \( v \) is component of the binary vector; \( \delta_r \) is the computational constraints for each resource, which is calculated by the computational load multiplied by the boundary of the task. The value of the computational constraints should be less than maximum ability of processing as indicated in the following expression \( \delta_r(\tau) < \delta_{r_{\text{max}}}(\tau) \). The latter component can be found using the process capability index, which represents the capability of the system to produce output within specification limits [3,4,5].

As the task consists of identifying and tracking of mobile agent, equation will serve as a mathematical model to calculate the maximum capability:

\[
C(\tau) = \frac{C_p}{\sqrt{1 + \frac{(\mu - \tau)^2}{\sigma^2}}}
\]

where \( C(\tau) \) is the Tagushi capability index and represents the maximum capability \( \delta_{r_{\text{max}}}(\tau) \) taking into consideration that the output is normally disturbed; \( C_p \) estimates what the process is capable of producing if the process mean were to be centred between the specification limits; \( \mu \) is the estimated mean of the process and \( \sigma \) is the standard deviation.
4. Discussion and results
By implementing the described architecture and necessary coding to a miniature UAV autopilot with the following resources 4GB 1600 GHz LPDD3, the transmission of the data could be achieved every 30ms. The generation and tracking of the trajectory is depicted in fig.3.

Additionally, it was taken into consideration the consumption of the electrical power needed to maintain the altitude of the flight. When the processing power increases, the power needed to maintain the process increases as well.

Figure 4 illustrates the performance of the UAV during agent tracking with different battery levels. It is clearly seen that the dropping 25% of the battery power will have effect on the performance on the flight stability. It was noticed that with full battery capacity and substantially functional system, the miniature UAV was able to achieve 15 minutes flight without registering real performance drop. Taking into consideration that the UAV speed can be within 3-5 m/s, the drop of power consumption while the functional degradation is active can decrease the flight range by 27%.

In light of the above, we can state that the functional degradation can be used for a limited period of time, which should be taken into consideration in equation (2). Hence, it is necessary to analyze Tagushi capability index with reference to the computational constraints and power. It is also important to state that the flight conditions were taken as ideal, where the range finder waves are not disturbed by humidity.
or facing wind, temperature is adequate not to impose additional heat dissipation on the rotors and processing units.

References

[1] Johnson T, Mitra S Handling failures in cyber-physical systems: potential directions. 2 pages Retrieved from: https://pdfs.semanticscholar.org/bc7d/8052cca42141087b70d9dca6d21662f333d6.pdf

[2] Glass M et al 2009 Incorporating graceful degradation into embedded system design. In Proceedings of design, automation and Test in Europe 320-323

[3] Mitchell R Process Capability indices. ASQ statistics division 18(1) 16-21.

[4] Vannman K 1995 A unified approach to capability indices Statistica Sinica 5 805-820.

[5] E. Kurekova 2001 Measurement process capability – trends and approaches Measurement Science Review 1(1) 43-46

[6] Jamshidi M et al. 2011 Cyber physical control of unmanned aerial vehicles. Scientia Inarica D 18(3) 663-668.

[7] Samad T, Annaswamy A. M. 2011 The impact of control technology. IEEE Control System Society, New York. Retrieved from: https://www.ieeecss.org

[8] Rajhans A et al 2009 An architectural approach to the design and analysis of cyber physical systems. 3rd International workshop on multi-paradigm modeling.

[9] Sayfeddine D, Bulgakov A, Kruglova T 2014 Dynamic Agent tracking using neural networks for an UAV [in Russ.], Inzhirnir Vestnik Dona. Retrieved from https://www.ivdon.ru/ru/magazine/archive/11y2014/2293

[10] Sayfeddine D 2009 UAV online agent tracking using geometric approach. International Scientific Conference 2 109-112.

[11] Adya M, Collopy F 1998 How effective are neural networks at forecasting and prediction? A review and evaluation Journal of forecasting 17 481-495.