

# Multi - level classification and formulation of an integration framework for estimation/ communication/ computation (EC<sup>2</sup>) co-design

Alireza Farhadi, Carlos Canudas de Wit

► **To cite this version:**

Alireza Farhadi, Carlos Canudas de Wit. Multi - level classification and formulation of an integration framework for estimation/ communication/ computation (EC<sup>2</sup>) co-design. [Research Report] Rapport de recherche. 2010, pp.1-39. hal-00536613

**HAL Id: hal-00536613**

**<https://hal.archives-ouvertes.fr/hal-00536613>**

Submitted on 18 Nov 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



## GRANT AGREEMENT N223866

Associate Deliverable	D06.01
Nature	Advanced Report

**D06.01 -Multi - Level Classification and Formulation of an Integration**

**Framework for Estimation/ Communication/ Computation (EC2) Co-design**

Report Preparation Date	25/JULY/2010 Project month:22
Authors	Alireza Farhadi, Carlos Canudas de Wit, INRIA, France
In Collaboration with:	Sandro Zampieri, Luca Schenato, UNIPD, Italy
Report Version	V1
Doc ID Code	INRIA-D06.01-25JULY2010-V1
Contract Start Date	01/SEP/2008
Duration	36 months
Project Coordinator :	Carlos CANUDAS DE WIT, INRIA, France



Theme 3:

Information and Communication Technologies

# Contents

<b>0</b>	<b>Executive summary</b>	<b>3</b>
<b>1</b>	<b>Fleet of underwater vehicles</b>	<b>5</b>
1.0	Introduction . . . . .	5
1.1	Control architecture . . . . .	5
1.2	Communication . . . . .	6
1.3	Computation . . . . .	9
1.4	Complexity/number of AUVs . . . . .	10
1.5	Energy . . . . .	10
1.6	C4E co-design framework for the fleet of AUVs . . . . .	11
<b>2</b>	<b>Smart networks of cameras for motion capture</b>	<b>12</b>
2.0	Introduction . . . . .	12
2.1	Control architecture . . . . .	14
2.2	Communication . . . . .	15
2.3	Computation . . . . .	15
2.4	Complexity . . . . .	15
2.5	Energy . . . . .	16
2.6	C3 co-design framework for the smart networks of cameras for motion capture . . . . .	16
<b>3</b>	<b>Smart networks of cameras for surveillance applications</b>	<b>17</b>
3.0	Introduction . . . . .	17
3.1	Control architecture . . . . .	19
3.2	Communication . . . . .	19
3.3	Computation . . . . .	19
3.4	Complexity . . . . .	20
3.5	Energy . . . . .	20
3.6	C4E - co-design framework for the smart networks of cameras for surveillance . . . . .	20
<b>4</b>	<b>Integration framework for EC2 co-design</b>	<b>21</b>
4.0	Introduction . . . . .	21
4.1	EC2 co-design framework for the fleet of AUVs . . . . .	22
4.2	EC2 co-design framework for the network of surveillance cameras	29
4.3	Existing EC2 results . . . . .	33
4.4	Potential aggregation of the existing results and the proposed problems . . . . .	37

## 0 Executive summary

This report is an overview of the research activities regarding the WP06 (C4E co-design) of the FeedNetBack European project for the first six months of the year 2010 at INRIA. The research team consists of Post Doctoral Fellow Alireza Farhadi and Director Carlos Canudas de Wit.

In preparing this report we had:

- A meeting with our industrial partner Ifremer.
- A three days visit at the University of Padova (UNIPD), Italy.
- A short discussion with our industrial partner Videotech.
- Long discussions with Sandro Zampieri and Luca Schenato from UNIPD.

We also received some results from our colleagues working in ETH (Swiss). The objective of the FeedNetBack project is to propose a co-design framework, which allows the integration of control-estimation, communication, computation, complexity, and energy in networked control systems. This co-design framework is developed for the following case studies:

- *A fleet of Autonomous Underwater Vehicles (AUVs)*
- *Intelligent camera networks for motion capture*
- *Surveillance systems using a network of smart cameras*

The three case studies have been selected to demonstrate the wide spectrum of possible applications of the FeedNetBack project: From systems with relatively few, highly mobile nodes, communicating over a network subject to communication imperfections; to systems with a very high number of immobile nodes, with high available bandwidth but also high computation requirements (smart camera network for surveillance applications and motion capture).

To create such a co-design framework we first need to fully understand the constraints imposed by control, communication, computation, complexity, and energy on the above case studies. This is the first objective of this report. The second objective is to formulate an estimation/ communication/ computation co-design framework which is applicable to the above case studies.

To achieve these goals, in Section 1, we study fleet of AUVs; and following

our discussions with Ifremer, we identify the interactions between control, communication, computation, etc. in this case study. In Section 2, we study smart networks of cameras for motion capture; and following our discussions with Sandro Zampieri and Luca Schenato, we identify the interactions between different components (control, communication, etc.). In Section 3, we study smart networks of cameras for surveillance applications; and following our discussions with Videotech, Sandro Zampieri and Luca Schenato, we identify the interactions between different components. Then, in Section 4, based on this studies, we formulate an integration framework for estimation/communication/ computation co-design which is applicable to fleet of AUVs and smart camera network for surveillance applications.

# 1 Fleet of underwater vehicles

## 1.0 Introduction

One of the case studies in the FeedNetBack project deals with a coordinated group of Autonomous Underwater Vehicles (AUVs) supervised by Autonomous Surface Vehicles (ASV). The objective is to detect and follow gradients of concentration of an underwater source flow (e.g., fresh water source flow) located in the bottom of ocean using sensed data provided by scientific sensors (payload sensors). This mission is called gradient referenced search mission which has a considerable potential for network control theory. The gradient referenced search mission is a cooperative fleet control mission. The aim of the cooperative fleet control law is to form a (uniformly distributed) circular formation of agents (AUVs) whose center and radius are given by a time-varying reference.

Each AUV is equipped with a powerful on-board computer and uses batteries with limited life time (typically between 5 to 10 hours) as the on-board power supply. It is also equipped to measure a level of concentration locally to determine a gradient of that concentration. The measured concentration data is used to produce the reference center and radius. To form a circular formation, AUVs also exchange their navigation data. The sensed concentration data and navigation data are exchanged via acoustic waves. The corresponding channel is subject to communication imperfections, e.g., short communication range, transmission delay, noise, fading, the Doppler shift effect, etc.

In the following subsections, we study the control architecture of the fleet; and the effects of communication imperfections, computation, complexity, and energy in the fleet control of AUVs. This study is used to understand the fundamental interactions between control, communication, computation, complexity and energy in the fleet of AUVs.

## 1.1 Control architecture

The fleet control law consists of a high level and a low level (local) control per each AUV. The high level control produces the time varying reference center and radius so that the fleet eventually reach the source flow. On the other hand, the low level control forms a (uniformly distributed) circular formation of AUVs with the center and radius given by the high level control.

## 1.2 Communication

The only practical way for underwater communication is via acoustic waves. Transmission via acoustic waves is subject to significant propagation loss, environmental noise, low speed (which results in propagation delay and the Doppler shift), and multi paths transmission, as described below:

**Propagation loss:** The propagation loss depends on the range and frequency. The Transmission Loss (TL in dB) increases with range ( $R$  in meter) and frequency ( $f$  in kHz) as described by:

$$TL = 20 \log(R) + \alpha \frac{R}{1000}, \quad (1)$$

where  $\alpha$  is the absorption loss which depends on frequency, temperature, and depth.

This loss significantly limits the bandwidth. The practical bandwidth for underwater communication is up to 30 kHz. For a bandwidth up to 30 kHz, the maximum communication range is normally limited to 1km; and for a bandwidth up to 10kHz, the maximum communication range is normally limited to 3km.

**Environmental noise:** The noise received by the receiver bandwidth limits the maximum acoustic range of the system. There are two kinds of noise: The ambient noise (e.g., hydrodynamics noise causes by wind, rain or biological sea-life noise) and the vehicle noise.

**Low speed transmission:** The nominal value of sound speed is 1500m/s. The low value of sound speed has two main effects: The propagation delay and the Doppler shift effect distortion, as described below:

*Propagation delay:* The propagation delay depends on the sound speed value and the range between acoustic transmitter and acoustic receiver. This delay is important for long distances and limits applications with temporal constraints.

*The Doppler shift:* The Doppler shift distortion corresponds to a frequency shift. It happens when acoustic systems (transmitter/receiver) are moving. It is described by:

$$f_r = \left( \frac{v + v_r}{v + v_s} \right) f_s, \quad (2)$$

where  $v$  is the velocity of acoustic wave,  $v_r$  is the velocity of receiver relative to the medium (it is positive if the receiver is moving towards the source),

$v_s$  is the velocity of the source relative to the medium (it is positive if the source is moving away from the receiver),  $f_r$  is the frequency at the receiver, and  $f_s$  is the frequency of the source.

**Multipaths:** The ocean surface varies from a glossy smooth reflector to a very rough and turbulent surface that scatters sound in a random fashion. The ocean bottom also has a wide variety of compositions, slopes, and toughness, all of which affect sound transmission. The effects of multi paths include constructive and destructive interferences and phase shifting of the signal which results in fading. The effects of multi paths are more present in shallow water and over long distance communication.

For underwater transmission of data, acoustic modems are used. These models convert digital data into special underwater sound signals produced by ultrasonic devices. These signals pass through underwater acoustic channels; and then are received by a second acoustic modem and converted back into digital data.

In the fleet of AUVs, two types of communication are possible: Between AUVs, and between AUVs and ASV. For communication between AUVs, omnidirectional transducers are used. The allocated bandwidth is 15-28 kHz, the maximum bit rate is 480 bits/s, and the communication range is limited to 1km. The error level for this range is normally 1 bit error over 10 transmitted bits. For communication between AUVs and ASV, omnidirectional looking upward transducers are used; and the allocated bandwidth is 9.5-13 kHz. The maximum bit rate for this transmission is 100 bits/s, the maximum communication range is 3 km, and the error level for this range is normally 1 bit in 10 transmitted bits. Although the available bandwidths for underwater communication are limited, they are enough for a communication without distortion.

Acoustic modems use different kind of data compression, coding, and correction techniques to increase the quality of acoustic communication. They also exchange modem data (e.g., measured noise level); and subsequently source can be adjusted accordingly. Our industrial partner (Ifremer) developed a packet erasure transmission technique with feedback acknowledgement for exchanging data. In this type of communication receiver knows if the received packet contains errors (using powerful error detection/correction techniques); and subsequently it disregards the packets containing errors. Also, transmitter knows (via feedback acknowledgment from receiver) that whether the transmitted packet was disregarded (or not) by receiver. Here, the erasure probability is unknown and varies with time (due to sudden changes in environment, multi path, etc). Typical value for erasure probability is something



between 0.2 and 0.5.

To form circular formation, the navigation data must be exchanged between AUVs. This may result in interference. To eliminate interference, we need to use suitable multiplexing scheme. Underwater communication is wide-band. Therefore, Orthogonal Frequency Division Multiplexing (OFDM) is very suitable for underwater communication. OFDM is a multi-carrier modulation technique, in which wideband data is transmitted via orthogonal sub-carriers. This method eliminates the interference. In OFDM, the transmitter and receiver must be synchronized (e.g., the receiver must know the frequency used for modulation). But, underwater communication is subject to the Doppler shift effect due to very low speed of underwater waves. Although the Doppler shift effect is small for underwater acoustic waves, it results in asynchronization between transmitter and receiver; and therefore significantly damages the quality of a communication which is based on OFDM. Consequently, we need to compensate the Doppler shift effect by estimating this parameter using test signals. Using the estimated value of the Doppler shift, the effects of this parameter in OFDM are reduced significantly.

Note that from the practical point of view, the available under water sound modems do not support OFDM techniques. In fact, up to now the only practical way for underwater multiple accessing without collision is via Time Division Multiple Access (TDMA) scheme. But, TDMA scheme avoids simultaneous transmission from multiple nodes. This results in latency which damages the quality of control. This latency grows up with the number of AUVs. Roughly speaking when the number of AUVs is more than ten, this latency is too long that the circular formation of AUVs is impossible to form. The low speed propagation of underwater sound also results in communication delay. The value of this delay depends on the range between acoustic transmitter and acoustic receiver. The value of communication delay can be also measured using test signals. Therefore, in the coordinated fleet of underwater vehicles, communication delay is a known parameter.

In a coordinated group of AUVs, for the safety reasons and also for global positioning, there is communication between AUVs and ASV. Also, from time to time, the collected data (images, etc) by AUVs are sent to ASV.

Therefore, in these applications, the payload data can be also sent from each AUV to ASV; and the reference center and radius can be produced at ASV and then broadcasted to all AUVs. As shown in [6] to form a uniformly distributed circular formation with a time varying center which eventually reaches to the source flow, the reference center and radius must be updated using the sampled concentration data of all AUVs. But, communication between two AUVs which are far from each other is subject to high imper-

fections (absorption, noise, etc); and therefore, is not possible. Hence, the payload data are sent to ASV; and the reference center and radius is updated there and then broadcasted to all AUVs. Note that when the fleet of AUVs are far from ASV, it might be necessary to position some AUVs between the fleet of AUVs and ASV to maintain communication between the fleet and ASV.

Using the above strategy, the payload data produced by each AUV are transmitted to ASV; and the reference center and radius are updated there and broadcasted to all AUVs. These communications are subject to transmission delays. However, the frequency of updating the reference center and radius does not need to be high; and therefore, the effects of these delays are negligible. Nevertheless, as mentioned earlier, for having a circular formation in 3D space, AUVs need to exchange navigation data via acoustic waves. Consequently, the communications between AUVs are subject to known time delays, in which the effects of these delays must be taken into account when we design the low level control laws.

**Effective communication constraints:** Above discussion reveals that in the fleet of AUVs, communication is subject to:

- 1) *Short communication range:* The communication range between AUVs is limited to 1km and between AUVs and ASV is limited to 3km.
- 2) *Noise:* Roughly speaking there is 1 bit error in 10 transmitted bits. Using error detection/correction techniques the communication channel can be modeled by a packet erasure channel with unknown and time varying erasure probability.
- 3) *Delay:* Underwater transmission is subject to communication delay which can be measured.
- 4) *Communication complexity:* Up to now, the only practical way for underwater multiple accessing without collision is via TDMA. TDMA results in communication complexity. When the number of AUVs increases, the latency due to TDMA may be too long that the circular formation is impossible to form.

### 1.3 Computation

Each AUV is equipped with a powerful on-board computer; and therefore the effects of computation on the low level control performance is negligible. However, each AUV sends its sampled concentration data to ASV; and ASV produces the reference center and radius; and then broadcasts the updated information to all AUVs. Due to long transmission delays in exchanging data between ASV and AUVs, the available time for processing the received in-

formation from AUVs and updating the reference center and radius by ASV on-board computer is limited. Besides, ASV on-board computer is responsible for other tasks. Therefore, high level control law must be developed in the presence of limited computational resource.

## 1.4 Complexity/number of AUVs

In the fleet of AUVs we need to employ a proper number of AUVs. AUVs are expensive vehicles (each AUV costs around 1.5 million euros). Therefore, it is desirable to accomplish gradient search mission by employing as minimum AUVs as possible. But, employing small number of AUVs results in a relatively poor sampled concentration data which increases the duration of mission. In fact, by using small number of AUVs, the search mission may be too long that is beyond the life time of AUVs.

Small number of AUVs also results in a poor control performance. However, small number of AUVs corresponds to low communication complexity. As discussed earlier, the only practical way for underwater multiple accessing without interference is via TDMA. But, the latency due to TDMA grows up with the number of AUVs. In particular, if the number of AUVs is large, the latency may be too long that the circular formulation is impossible to form. Thus, the number of AUVs affects:

- 1) Cost of mission (small number of AUV = relatively cheap mission).
- 2) Duration of the search mission (small number of AUVs= long mission).
- 3) Quality of control (small number of AUVs = poor control performance).
- 4) Communication complexity (small number of AUVs = low communication complexity).

Therefore, in the fleet of AUVs we need to employ a proper number of AUVs to have a reasonable communication complexity, good quality of control, and a suitable duration for the search mission.

## 1.5 Energy

The energy consumption of on-board computer, sensors, and communication transducers of each AUV are all negligible compared with the energy consumption of propulsion devices which consume most of energy. Therefore, the life time of AUV does not significantly increase by designing communication/computation aware energy techniques. Therefore, in the fleet of AUVs we need to employ enough number of vehicles to accomplish the search mission before finishing the available on-board energy resources by the propulsion devices.

Interacting component	Effective on
Control	Communication
Communication	High level control and low level control
Computation	High level control
Complexity	Low level control, communication, computation, and energy
Energy	Complexity

Table 1: Summary of interactions between control, communication, computation, complexity, and energy in the fleet of AUVs

## 1.6 C4E co-design framework for the fleet of AUVs

As explained earlier, the fleet control of AUVs involves a two-level control: Low level control and high level control. Low level control forms a circular formation. But, to have a circular formation, each AUV needs to know other AUVs navigation data. This requirement results in employing a TDMA based communication technique. Also, communication in the coordinated group of AUVs is subject to short range, (known) communication delays, and noise/packet dropout. Therefore, these effects must be taken into account when designing fleet control laws. Moreover, limited computational resources affect the high level control design. Moreover, the number of AUVs affects the cost of mission, quality of control, communication complexity, required computational load associated with high level control, and duration of mission. Consequently, for a given control and communication strategy, we need to find a proper number of AUVs so that we have a reasonable communication complexity, good quality of control, and a proper duration for the search mission. Each AUV also has a limited power supply which affects the number of vehicles involved in the search mission. In the above table we summarized the interactions between control, communication, computation, complexity, and energy (C4E) in the fleet of AUVs.

## 2 Smart networks of cameras for motion capture

### 2.0 Introduction

One of the case studies in the FeedNetBack project deals with a network of smart cameras for motion capture. Depending on the application, motion capture systems use a few to hundreds of video cameras to capture the motions of physical objects (actors) and translate them into 3D models of the objects. The captured motion is presented in 2D image planes of cameras and translated into 3D trajectories in space in a central PC by a suitable reconstruction algorithms. This is often used to create realistic animations of fictional characters. In these systems reflective markers are placed on the actors' bodies, which are tracked on the 2D image planes of the cameras. In traditional optical motion capture systems a number of cameras surround the capture volume with the requirement that two or more cameras must observe any given marker at all times in order to reconstruct the marker position in space. Each optical unit consists of a distinct video camera, a strobe head unit, a suitable lens, optical filter, and cables. The video camera assembly is a highly advanced unit containing a digital sensor of up to 16 megapixels, on-board FPGA (field programmable gate array) and DSP (digital signal processing unit) and a gigabit Ethernet connection. The cameras operate on a dedicated megabit network (a 100 Base-T Ethernet) for synchronization and data transfer to the central PC.

The fixed nature of traditional motion capture cameras cannot adjust dynamically to accommodate the motion of the observed objects. The quality of the 3D reconstruction will be improved significantly if the system also uses Pan - Tilt -Zoom (PTZ) cameras (which can rotate horizontally and vertically) by focusing on the objects needing higher details (e.g., hands or face). PTZ cameras also increase the volume of the physical space that actors can move through; while being motion captured.

Each PTZ camera is connected via an analog link to an advanced processing unit called Video Agent (VA), which has on-board DSP and Ethernet 10/100 Mbps connection. This unit collects raw videos and has real-time video processing capabilities. Video agents are used to remotely actuate/control PTZ cameras. Video agents can directly communicate to the central PC via 100/1000 Mbps LAN (local area network). They send information data and compressed videos to the central PC, whenever is required.

In general, there are two distinguishable modes in motion capture systems: Calibration mode and operational mode, as described below:

**Camera Calibration Mode:** Camera calibration is the process of determining the approximate pose of all the motion capture cameras as well as their internal optical parameters. The pose of a camera is the translation and orientation of the camera in a reference coordinate system which is used by the central PC for reconstruction in 3D. For a motion capture system with fixed cameras, calibration step consists of capturing a trial where the cameras observe a calibration wand with five markers mounted on it in known relative positions.

Calibration for a system consisting of a mixture of fixed cameras and cameras mounted on PTZ heads is very similar to the traditional setup. A subset of the connected cameras will be mounted on pan - tilt -zoom heads which can be controlled from the central PC. The standard calibration step will be performed with the PTZ cameras pointing towards the center of the volume and not moving. As a result, the internal parameters and initial pose for all cameras are determined.

In this system, in addition of calibration wand, some markers are placed in unknown positions around the volume. These markers are called beacon markers and will be used for finalizing the calibration of PTZ cameras, as described below:

The PTZ controller has a built in encoder that reports an approximate pose for the camera in a reference frame determined by the PTZ controller software. In order for this pose estimate to be useful we need to produce a mapping of PTZ encoder pose to calibrated camera pose in the reference coordinate system. This pose mapping will be determined automatically after the standard calibration has been performed. The calibration software will exercise the PTZ cameras through their entire motion range with the restriction that they keep a number of the beacon markers in view. This will produce a mapping that can convert any encoder pose into an approximate calibrated pose.

We now describe the second phase which is the operational mode.

**Operational Mode:** In this mode, distributed cameras capture the motions of physical actors and send the captured data to the central PC. The central PC then translate this information into 3D models of actors. The control systems developed for this mode will direct the PTZ cameras to focus on areas of interest and thereby improve the effectiveness of camera resources.

In the following subsections, we study this case study in more details by describing its control architecture and the effects of computation and complexity. Subsequently, we identify the interactions between control, compu-

tation, and complexity.

## 2.1 Control architecture

The control system is divided into two parts: A local controller per camera and a central controller (running on the central PC) to provide oversight. Each of the PTZ cameras has a local controller. For controlling purposes, system supplies two communication routes: There is a Local-Central communication path for general use which allows the local controllers to communicate information to the central controller. There is also a Central-Local communication path for communication to the local controllers. The two classes of controllers have different information available and different objectives, as described next.

**Central Controller:** The central controller has all known information available such as 3D points and labels (pre-determined identifiers for specific markers on objects), camera poses, 2D to 3D associations and 2D tracks. The objective of the central controller is to direct local controllers to the areas of interests. Central controller also makes decisions as to which PTZ camera to assign to which object, or to multiple objects if multiple objects are visible from a single PTZ camera. The performance of the camera control algorithms is formulated as the minimization of an error metric based on the reconstructions produced. The reconstruction algorithm produces a reconstruction and associated covariance matrix for each visible marker. The performance matrix can be then calculated by

$$E = \sum_{i=1}^m \min(\text{trac}(C_i), \text{threshold}),$$

where  $C_i$  is the covariance matrix for each of the  $m$  reconstructions and threshold is the maximum error per reconstruction which will be counted if the reconstruction is missing. This error is minimized when all markers are viewed by all cameras. Therefore, the central controller makes decision for PTZ cameras so that markers are viewed by as many cameras as possible.

**Local Controller:** The local controllers are supplied with some static information such as camera internal parameters and mapping of PTZ controller encoder poses to calibrated camera pose frame. For each frame of data observed by the camera, the controller will receive 2D coordinates as well as temporal associations. Additionally, the controller will receive information from the central controller for a frame in the past with fixed latency of 50ms.



This information includes 3D coordinates, 2D to 3D associations and all known camera poses at that time. The local controller's objective would be to follow markers specified by the central controller in the presence of i) Clutter appearing due to background reflections, ii) Processing delays in centrally computing 3D information and camera poses (typically 50 ms)

## 2.2 Communication

As mentioned earlier, for controlling purposes, system supplies two communication routes: There is a Local-Central communication route for general use which allows the local controllers to communicate information to the central controller. There is also a Central- Local communication route for communication to the local controllers. The communication occurs on a dedicated megabit network which guarantees low latency and no packet loss. The latency of data being captured by the camera until being received by the central PC is less than 5ms. The latency from data capture to processing by the DSP on-board the camera is significantly less. In fact, the temporal and spacial uncertainties such as packet delays, bandwidth limitations, etc. are eliminated by this dedicated network.

## 2.3 Computation

Only a limited subset of information such as 2D positions is available directly on the camera; while the central controller has more information such as 3D marker positions and labels. Therefore, this information must be communicated from the central controller to the local controllers. But, as pointed out, there are processing delays in centrally computing 3D information and camera poses. Consequently, the local controller will receive information from the central controller for a frame produced almost 50ms ago. That is, there is a computational effect that results in a fixed latency of 50ms. This effect must be considered in the design of local controllers.

## 2.4 Complexity

Motion capture systems normally involve between 2 and 300 cameras. When the number of cameras increases, the centralized process of reconstructing the 3D positions of markers demands a suitable scalability technique (e.g., distributed computation/control techniques) to reduce high computational demands and control complexity.



Interacting component	Effective on
Computation	Control
Complexity	Control and computation

Table 2: Summary of interactions between control, computation, and complexity in the smart networks of cameras for motion capture

## 2.5 Energy

The cameras use power-over Ethernet to power cameras; and thus the issue of energy consumption for cameras is secondary. Similarly, the central PC is supplied by the city power line; and therefore the issue of energy consumption for the central PC is also secondary.

## 2.6 C3 co-design framework for the smart networks of cameras for motion capture

Above discussions reveal that in the smart networks of cameras for motion capture, there are interactions between control, computation, and complexity (for systems with many cameras). There is a processing delay in centrally computing 3D information and camera poses. Consequently, the local controller will receive information from the central controller with a fixed latency. Moreover, when the number of cameras increases, the centralized process of reconstructing the 3D positions of markers demands a suitable scalability technique to reduce high computational demands and control complexity. In the above table we summarized these interactions.

## 3 Smart networks of cameras for surveillance applications

### 3.0 Introduction

In intelligent systems for outdoor video surveillance, many PTZ cameras with on-board video processing capabilities are distributed through a site and connected via a LAN network. The volume to be monitored is generally large. The goal is to cooperatively detect and track objects by cameras, as they move through the site.

Target detection involves scanning the monitored site which is called patrolling. After detecting an object, each camera pans and tilts such that an object detected in the assigned camera is in the vantage position in the camera's image plane; and consequently its images is captured. When a PTZ camera detects an object it can also focus on such an object and tracks it at a higher level of zoom to gather finer details.

Each PTZ camera is equipped with a Video Agent (VA), where a VA is a logical model of a network-connected processing unit with an active PTZ camera. Video agents are connected to each other and several personal computers via a 100 Mbps Ethernet cable LAN. Video agents are used to actuate/control PTZ cameras. They can directly communicate to each other without the need to communicate to a central server; and can send information data and compressed videos to other video agents and one or more remote personal computers in real time, whenever is required.

In general, there are two distinguishable modes in smart networks of cameras for surveillance applications: Calibration mode and cooperative detection/tracking mode, as described below:

**Calibration Mode:** The calibration process provides the system with the information required for performing detection and tracking tasks. In this process, each camera learns which are its neighboring cameras and which areas are visible by them. This process also provides knowledge about relationships between fields of view of neighboring cameras as a function of the PTZ states.

For each camera, when a target is detected by a camera, its coordinates are expressed in terms of camera pan and tilt parameters and the target's estimated depth and position in the camera frame. Consequently, if the target size is known, zoom can be related to the targets' distance from the camera image plane. Also, if the target is framed in the center of the camera image plane, pan and tilt can be related to the target position.

Given two cameras A and B, the calibration goal is to estimate a function that maps PTZ states in camera A to PTZ states in camera B when both cameras are watching the same object. In a real scenario, the reference object is in a given location of the monitored site. PTZ cameras are exploring their global field of view to detect the reference object. After a while; all PTZ cameras that can watch the reference object in that location reach a PTZ state such that the object is in the center of the corresponding image plane with a proper size. Consequently, it is possible to establish a correspondence between a PTZ state of a given camera and a PTZ state of other cameras watching the same object. Then, the reference object starts moving and more correspondence are established. The calibration mode can also produce the translation and orientation of the cameras in a reference (world) coordinate system. More detail on this issue can be found in [3].

**Cooperative Detection/Tracking:** Cooperative detection/tracking is a coordinated search of anomalous events. Detection is obtained by moving the field of views properly to maximize the overall probability of detection. When cameras perform PTZ movements independently, their field of view may overlap. Intuitively, overlapped field of views must be avoided to cover a wider area at a given time. Moreover, when multiple cameras focus on a same event, wide areas are no longer monitored; and other anomalous events will not be detected.

Due to the lack of centralized infrastructure, detection/tracking must be done in a distributed way using local information and communication to achieve global optimal performance. Nevertheless, it is difficult to achieve global optimal coordination starting from local cooperation.

When a moving object is detected, it must be followed by at least one camera. Specifically, the object must be remain inside the field of view of at least one camera as much as possible at the maximum zoom.

During tracking process, the 3D position, velocities, and orientation of the target is estimated, and this information is used by the local controller in order to follow the target as it moves inside the whole range of views of the camera under control. Also, when a target is leaving to a neighboring camera field of view, the neighboring camera must rush on the target to keep tracking it. It is the local controller's task to predict if a target is going to exist from its field of view and to alert an available neighboring camera.

Note that in order to measure the quality of tracking, several performance metrics can be associated with the tracking task such as:

- 1) The quality of information acquired from a target which can be measured in terms of:
  - i) Target dimension on the image plane of a camera.

- ii) Number of cameras focusing on a target at the same time.
- 2) The trace reconstruction, which is the number of frames that a target is effectively and continuously tracked by one or more cameras before being lost.

In the following subsections we study the control architecture of the smart networks of cameras for surveillance applications. We also study the effects of communication, computation, complexity, and energy on such networks. Similar to previous sections, this study is used to understand the fundamental interactions between control, communication, computation, complexity, and energy in the smart networks of cameras for surveillance applications.

### 3.1 Control architecture

The control system consists of one local controller, or video agent, per camera. The video agent is a plug - in point to supply control algorithms. Video agents are capable of acting and communicating to each other in a decentralized fashion. In other words, not all video agents have or need the same information, nor the same task. They can autonomously perform a task or cooperate with other agents. Note that camera PTZ controllers cannot move arbitrary fast. This has to be taken into account when designing control algorithms.

### 3.2 Communication

Currently, in the smart networks of cameras for surveillance applications, the communication occurs on a dedicated megabit cable Ethernet network which guarantees low latency and packet loss for communication between nearby VAs. However, our industrial partner (Videotech) is interested to use wireless networks for communication in near future. In this type of communication, transmission is subject to noise, short communication range, and limited bandwidth (if VA is equipped with a limited power supply), in which these effects must be considered in the development of control algorithms.

### 3.3 Computation

One of the important issues in the smart networks of cameras for surveillance is the issue of control aware computing. Camera platform is based on FPGA and DSP on-board processors that have slow development cycles and limited computing flexibility. This results in computational latency which needs to be considered when we develop the control algorithms.

### 3.4 Complexity

When the number of cameras or targets increases, the information to be processed and the complexity of the algorithms need to be managed properly. This is done by implementing a decentralized cooperative control technique, in which not all video agents have or need the same information, nor the same task.

### 3.5 Energy

Currently, our industrial partner is using cable Ethernet to interconnect VAs. Therefore, cameras use power - over Ethernet as the source of energy. Consequently, the issue of energy consumption is secondary. Nevertheless, in near future, our industrial partner is interested to use wireless communication and solar panels/ batteries as the source of energy. Consequently, it may be necessary to develop communication/computation aware energy techniques.

### 3.6 C4E - co-design framework for the smart networks of cameras for surveillance

Above discussions reveal that in the smart networks of cameras for surveillance applications, there are interactions between control, communication, computation, complexity, energy. Control algorithms require an instantaneous communication. Therefore, the commonly used data communication techniques are not suitable because they induce long decoding delays. Also, if we use wireless communication networks, we need to consider the effects of communication imperfections in the development of the control algorithms. Moreover, the smart networks of cameras for surveillance are subject to computational constraints; and therefore, the effects of these constraints must be taken into account when we design control/communication algorithms. Furthermore, when the number of cameras or targets increases, the complexity of control/communication algorithms must be managed properly. Finally, if cameras are supplied by solar panels/ batteries, we may need to develop communication/computation aware energy techniques.

In the following tables we summarized the interactions between control, communication, computation, complexity, and energy in the smart networks of cameras for surveillance applications.

Interacting component	Effective on
Control	Communication
Computation	Control, communication
Complexity	Control, communication, computation

Table 3: Summary of interactions between control, communication, computation, and complexity in the smart networks of cameras for surveillance applications - The case with cable Ethernet

Interacting component	Effective on
Communication	Control
Control	Communication
Computation	Control, communication
Complexity	Control, communication, computation
Energy	Communication, computation

Table 4: Summary of interactions between control, communication, computation, complexity, and energy in the smart networks of cameras for surveillance applications - The case with wireless links and solar panels/batteries

## 4 Integration framework for EC2 co-design

### 4.0 Introduction

As discussed in Section 1.3, each AUV sends its sampled concentration data via noisy channels to ASV. ASV on-board computer is responsible for reconstruction/estimation of the received sampled concentration data; and subsequently, updating the reference center and radius based on the estimated values. Due to long transmission delays in exchanging data between ASV and AUVs, the available time for processing the received information from AUVs and updating the reference center and radius by ASV on-board computer is limited. Besides, ASV on-board computer is responsible for other tasks. Therefore, high level control law, which is responsible for updating the reference center and radius, must be developed in the presence of limited computation resource and communication imperfections (e.g., channel noise).

Moreover, as discussed in Section 3.3, one of the important issues in the smart networks of cameras for surveillance applications, is the issue of control-estimation aware computing. Camera platform is based on FPGA and DSP on-board processors that have slow development cycles and limited computing flexibility. This results in computation latency which needs to be con-

sidered when we develop the tracking algorithms. These algorithms involve remote estimation over communication links. Thus, in the smart networks of surveillance cameras, coding algorithms for tracking must be developed in the presence of computation and communication constraints.

In the following section, we formulate an integration framework for Estimation/ Communication/ Computation (EC2) co-design for the fleet of AUVs and smart networks of cameras for surveillance applications.

#### 4.1 EC2 co-design framework for the fleet of AUVs

As discussed in Section 1, the objective of the gradient referenced search mission is to detect and follow gradients of the concentration of an underwater source flow using the sensed concentration data provided by the payload sensors of AUVs. This objective is achieved by employing a cooperative control law which consists of two levels: Low level control per each AUV and high level control located at ASV, which is responsible for updating the reference center and radius using the sampled concentration data. In the beginning of the search mission, AUVs form a uniformly distributed circular formation. Then, as shown in Fig. 1, the sampled concentration data  $\rho_i, \rho_j, \rho_n, \dots$ , which are obtained by AUVs, are transmitted via noisy channels to ASV. Then, ASV on-board computer updates the reference center  $C_d$  and radius  $R$  for the next circular formation by estimation/reconstruction of the transmitted sampled concentration data.

In the stationary regime, the diffusion law gives an exponential decreasing spatial distribution of concentration, represented in 2D by the following equation:

$$\rho(x, y) = \alpha e^{-K\left((x-x_s)^2+(y-y_s)^2\right)}, \quad (3)$$

where  $(x_s, y_s)$  is the center of the source,  $K$  the distribution spread, and  $\alpha$  the scaling coefficient. Note that the parameters  $\alpha$ ,  $K$ , and  $(x_s, y_s)$  are not known to ASV a priori.

Most of the time, above concentration distribution is distorted, e.g., due to waves. To represent this effect, above distribution is replaced by the summation of three concentric ellipses, as follows:

$$\begin{aligned} \rho(x, y) = & \alpha_1 e^{-\left(a_{11}(x-x_s)^2+a_{12}(x-x_s)(y-y_s)+a_{13}(y-y_s)^2\right)} \\ & \alpha_2 e^{-\left(a_{21}(x-x_s)^2+a_{22}(x-x_s)(y-y_s)+a_{23}(y-y_s)^2\right)} \\ & \alpha_3 e^{-\left(a_{31}(x-x_s)^2+a_{32}(x-x_s)(y-y_s)+a_{33}(y-y_s)^2\right)}, \end{aligned}$$

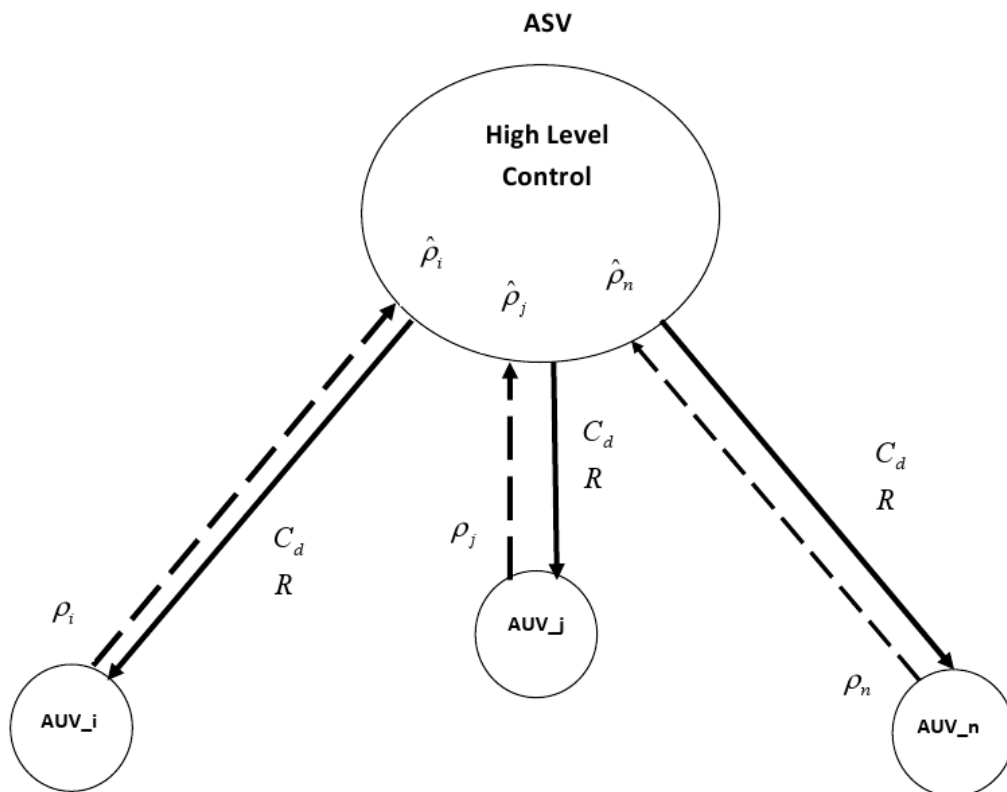


Figure 1: Exchanging the sampled concentration data and the reference center and radius between AUVs and ASV



where  $a_{11}a_{13} - \frac{a_{12}^2}{4} > 0$ ,  $a_{21}a_{23} - \frac{a_{22}^2}{4} > 0$ ,  $a_{31}a_{33} - \frac{a_{32}^2}{4} > 0$ , parameters  $a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}, a_{31}, a_{32}, a_{33}$ ,  $\alpha_1, \alpha_2, \alpha_3$  are unknown and subject to change, and the pair  $(x_s, y_s)$  is unknown but fixed.

For simplicity, in what follows we assume that the environment is calm enough such that the concentration distribution in 2D space is represented by the following elliptic model:

$$\rho(x, y) = \alpha e^{-\left(a(x-x_s)^2 + b(x-x_s)(y-y_s) + c(y-y_s)^2\right)}, \quad (4)$$

where  $ac - \frac{b^2}{4} > 0$ , the concentration parameters  $\alpha, a, b$ , and  $c$  are unknown and subject to slow change, and the pair  $(x_s, y_s)$  is unknown and fixed.

In the beginning of the search mission, AUVs form a uniformly distributed circular formation with center  $C_d$  and radius  $R$ . Arranz, Seuret, and Canudas de Wit in [4] presented policies for local controllers that result in uniformly distributed circular formation with center  $C_d$  and radius  $R$ . Now, consider the  $n$ th AUV ( $1 \leq n \leq N$ ) on a uniformly distributed circular formation of AUVs with radius  $R$  and center  $C_d$ , as shown in Fig. 2. Let the vector  $V_n$  (m/s) denote the vehicle's forward velocity; and  $\theta_n(t) = \omega t$  denote the heading angle, where  $\omega$  (rad/s) is the rotational speed (See Fig. 2). Note that  $|V_n|, R, \omega, C_d$ , and  $\gamma$  are known to ASV a priori, where  $|\cdot|$  denotes the magnitude. It is easy to verify that

$$\begin{pmatrix} x_n(t) \\ y_n(t) \end{pmatrix} = \begin{pmatrix} C_d \cos \gamma \\ C_d \sin \gamma \end{pmatrix} + \begin{pmatrix} R \cos(\omega t + \frac{2\pi}{N}n) \\ R \sin(\omega t + \frac{2\pi}{N}n) \end{pmatrix}. \quad (5)$$

$$\begin{pmatrix} \dot{x}_n(t) \\ \dot{y}_n(t) \end{pmatrix} = \begin{pmatrix} -|V_n| \sin(\omega t + \frac{2\pi}{N}n) \\ |V_n| \cos(\omega t + \frac{2\pi}{N}n) \end{pmatrix}. \quad (6)$$

Let  $\rho(x_n(t), y_n(t))$  be the sampled concentration data provided by the  $n$ th AUV at point  $(x_n(t), y_n(t))$ . Then,

$$\dot{\rho}(x_n(t), y_n(t)) = \frac{d\rho(x_n(t), y_n(t))}{dt} = \frac{\partial \rho(x_n(t), y_n(t))}{\partial x_n(t)} \frac{dx_n(t)}{dt} + \frac{\partial \rho(x_n(t), y_n(t))}{\partial y_n(t)} \frac{dy_n(t)}{dt}. \quad (7)$$

Note that above equation also involves the following terms:  $\frac{\partial \rho(x_n(t), y_n(t))}{\partial \alpha} \frac{d\alpha}{dt}$ ,  $\frac{\partial \rho(x_n(t), y_n(t))}{\partial a} \frac{da}{dt}$ ,  $\frac{\partial \rho(x_n(t), y_n(t))}{\partial b} \frac{db}{dt}$ ,  $\frac{\partial \rho(x_n(t), y_n(t))}{\partial c} \frac{dc}{dt}$ , and  $\frac{\partial \rho(x_n(t), y_n(t))}{\partial d} \frac{dd}{dt}$ . But, as

$$\frac{\partial \rho(x_n(t), y_n(t))}{\partial \alpha}, \frac{\partial \rho(x_n(t), y_n(t))}{\partial a}, \frac{\partial \rho(x_n(t), y_n(t))}{\partial b}, \frac{\partial \rho(x_n(t), y_n(t))}{\partial c}, \frac{\partial \rho(x_n(t), y_n(t))}{\partial d}$$

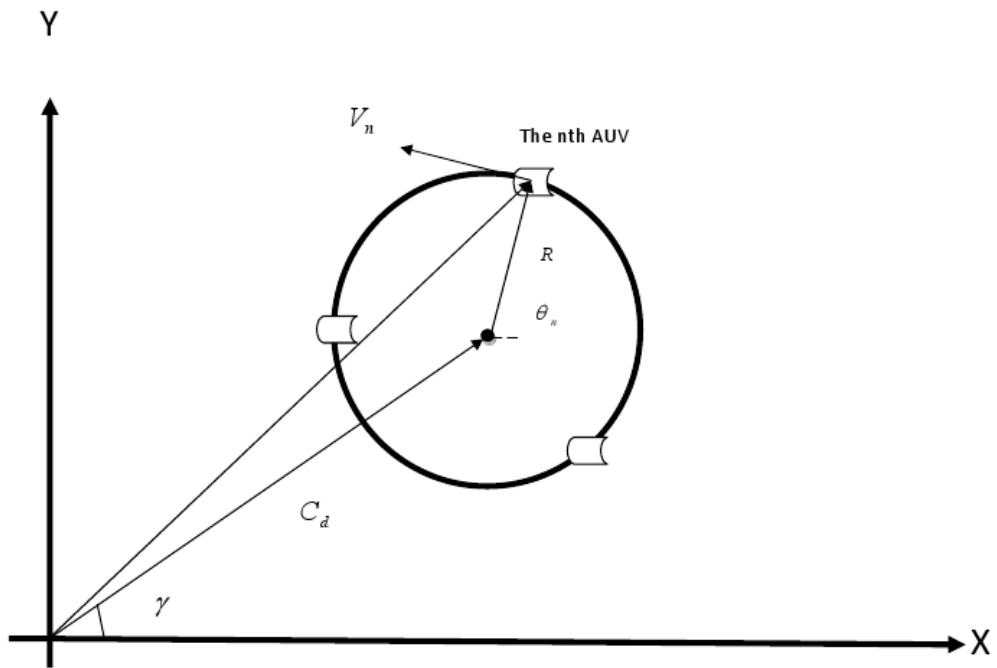


Figure 2: Illustration of an AUV on a circular formation with center  $C_d$  and radius  $R$

are bounded and the concentration parameters  $\alpha, a, b, c$  are changing slowly during AUVs maneuver on a given circular formation, we can assume that these terms are negligible.

From (4), (5), and (7), it follows that:

$$\begin{aligned}
 & \dot{\rho}(x_n(t), y_n(t)) \\
 &= [-\left(2a(x_n(t) - x_s) + b(y_n(t) - y_s)\right)\dot{x}_n(t) \\
 &\quad - \left(b(x_n(t) - x_s) + 2c(y_n(t) - y_s)\right)\dot{y}_n(t)]\rho(x_n(t), y_n(t)) \\
 &= \left[\left(2a(C_d \cos \gamma + R \cos(\omega t + \frac{2\pi}{N}n) - x_s) \right. \right. \\
 &\quad \left. \left. + b(C_d \sin \gamma + R \sin(\omega t + \frac{2\pi}{N}n) - y_s)\right)|V_n| \sin(\omega t + \frac{2\pi}{N}n) \right. \\
 &\quad \left. - \left(b(C_d \cos \gamma + R \cos(\omega t + \frac{2\pi}{N}n) - x_s) \right. \right. \\
 &\quad \left. \left. + 2c(C_d \sin \gamma + R \sin(\omega t + \frac{2\pi}{N}n) - y_s)\right)|V_n| \cos(\omega t + \frac{2\pi}{N}n)\right]\rho(x_n(t), y_n(t)).
 \end{aligned}$$

For the simplicity of notation, let us denote  $\rho(x_n(t), y_n(t))$  by  $\rho_n(t)$ . Then, the discrete version of the above linear time varying continuous dynamic system is:

$$\begin{aligned}
 & \frac{\rho_n(t+1) - \rho_n(t)}{T} \\
 &= \left[\left(2a(C_d \cos \gamma + R \cos(\omega T t + \frac{2\pi}{N}n) - x_s) \right. \right. \\
 &\quad \left. \left. + b(C_d \sin \gamma + R \sin(\omega T t + \frac{2\pi}{N}n) - y_s)\right)|V_n| \sin(\omega T t + \frac{2\pi}{N}n) \right. \\
 &\quad \left. - \left(b(C_d \cos \gamma + R \cos(\omega T t + \frac{2\pi}{N}n) - x_s) \right. \right. \\
 &\quad \left. \left. + 2c(C_d \sin \gamma + R \sin(\omega T t + \frac{2\pi}{N}n) - y_s)\right)|V_n| \cos(\omega T t + \frac{2\pi}{N}n)\right]\rho_n(t),
 \end{aligned}$$

where  $T$  is the sufficiently small fixed sampling time.

From the above equation we have the following discrete time linear time

varying dynamic system for the sampled concentration data:

$$\begin{aligned}
& \rho_n(t+1) \\
= & \left[ 1 + T \left( 2a \left( C_d \cos \gamma + R \cos \left( \omega T t + \frac{2\pi}{N} n \right) - x_s \right) \right. \right. \\
& \left. \left. + b \left( C_d \sin \gamma + R \sin \left( \omega T t + \frac{2\pi}{N} n \right) - y_s \right) \right) |V_n| \sin \left( \omega T t + \frac{2\pi}{N} n \right) \right. \\
& \left. - T \left( b \left( C_d \cos \gamma + R \cos \left( \omega T t + \frac{2\pi}{N} n \right) - x_s \right) \right. \right. \\
& \left. \left. + 2c \left( C_d \sin \gamma + R \sin \left( \omega T t + \frac{2\pi}{N} n \right) - y_s \right) \right) |V_n| \cos \left( \omega T t + \frac{2\pi}{N} n \right) \right] \rho_n(t) \quad (8)
\end{aligned}$$

That is, the discrete time system that can be used in ASV to estimate the sampled concentration data is of the following form:

$$\begin{aligned}
\rho_n(t+1) &= f_n(t, a, b, c, x_s, y_s, \omega, V_n, R, C_d) \rho_n(t) \\
z_n(t) &= \rho_n(t), \quad (9)
\end{aligned}$$

where  $z_n(t)$  is the observation signal, the function  $f_n(t, a, b, c, x_s, y_s, \omega, V_n, R, C_d)$  is a nonlinear continuous function of its arguments (and is defined from (8)), parameters  $a, b, c$  are not known and they are subject to slow change, and the pair  $(x_s, y_s)$  is not known but is fixed. It is also reasonable to assume that  $a_{min} \leq a \leq a_{max}$ ,  $b_{min} \leq b \leq b_{max}$ ,  $c_{min} \leq c \leq c_{max}$ ,  $x_{min} \leq x_s \leq x_{max}$ , and  $y_{min} \leq y_s \leq y_{max}$  with known  $a_{min}$ ,  $a_{max}$ ,  $b_{min}$ ,  $b_{max}$ ,  $c_{min}$ ,  $c_{max}$ ,  $x_{min}$ ,  $x_{max}$ ,  $y_{min}$ , and  $y_{max}$ . Note that in (9), the parameters  $\omega, V_n, R, C_d$  are known.

Fig. 3 illustrates the communication system between the  $n$ th AUV and ASV. As shown in Fig. 3, the sampled concentration data of the  $n$ th AUV (i.e.,  $z_n(t)$ ) is encoded to a string of binaries denoted here by  $\delta_n(t)$  and transmitted via a noisy channel to ASV. The noisy channel is modeled by the Binary Symmetric Channel (BSC). This channel flips the transmitted bit with the cross over probability  $0 \leq p_n \leq \frac{1}{2}$ . At ASV, the decoder receives the corrupted bits denoted by  $\bar{\delta}_n(t)$  and outputs  $\hat{\rho}_n(t)$ , which is the estimated version of the sampled concentration data  $\rho_n(t)$ .

As discussed in Section 1, this transmission is subject to long transmission delay; but, this delay is known a priori. Therefore, without loss of generality, we do not need to consider this delay in the development/analysis of the coding algorithms (the receiver knows that the decoder output,  $\hat{\rho}_n(t)$ , corresponds to the sampled concentration data  $\rho_n(t - t_d)$ , where  $t_d$  is the known transmission delay).

Since ASV is equipped with high power supplies, the power of the transmitter of ASV can be chosen high enough such that the communication

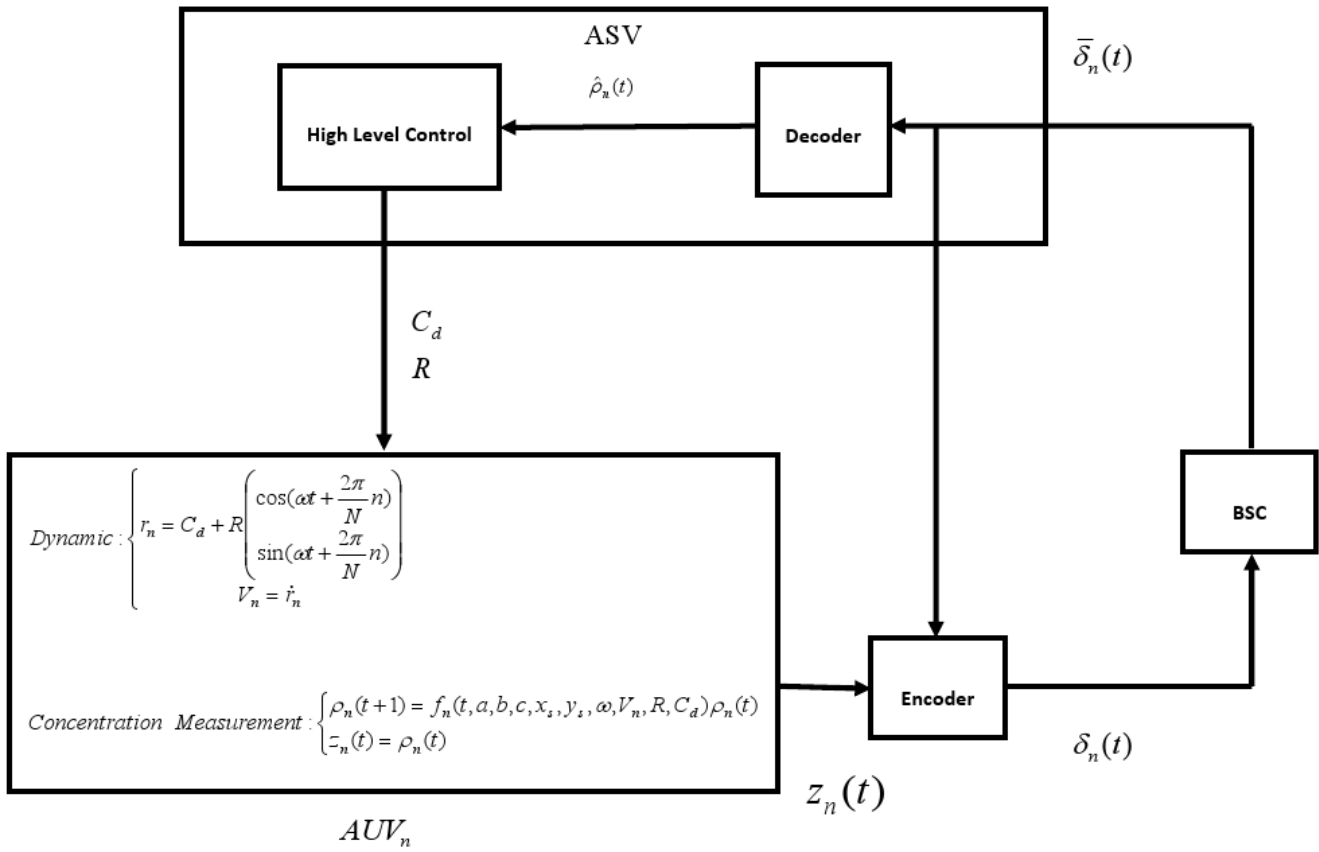


Figure 3: Communication system between the  $n$ th AUV and ASV

from ASV to AUVs is noiseless. Therefore, when ASV updates the reference center and radius for the next circular formation, it broadcasts them to all AUVs noiseless. For the same reason, in the communication system of Fig. 3, a noiseless feedback acknowledgment from the receiver to transmitter is possible. However, this feedback may result in a longer search mission due to long transmission delay in communication from ASV to AUVs. For the same reason (long transmission delays in communication between ASV and AUVs), the available time for processing the information received from AUVs and updating the reference center and radius is limited. In other words, implementing computationally expensive algorithms, which result in high computation latency, can result in a very long search mission so that the batteries of AUVs may die before reaching to the source flow.

In [6], a high level algorithm for forming a uniformly distributed circular formation of AUVs with a time varying center, which eventually reaches to the source flow, was presented. The algorithm works if a) The concentration distribution has a maximum at some point and compact elliptical level set, and b) The reference center is updated using the exact sampled concentration data of all AUVs (i.e.,  $\hat{\rho}_n(t) = \rho_n(t)$ ,  $\forall n$ ). Therefore, to implement this algorithm, we need to develop a coding algorithm (via designing proper encoder and decoder for the communication system of Fig. 3) that results in an accurate estimation (i.e.,  $\hat{\rho}_n(t) \approx \rho_n(t)$ ,  $\forall t \geq t_a$ ) as fast as possible in the presence of channel noise, computation constraint, and unknown concentration parameters. This is the problem that we are interested to solve for the fleet of AUVs.

## 4.2 EC2 co-design framework for the network of surveillance cameras

As discussed in Section 3.3, tracking is one of the important tasks in the smart networks of cameras for surveillance applications. During tracking process, the 3D position, orientation, and velocities of the target is estimated (for instance, using the extended Kalman filter); and this estimation is used by the local controller in order to follow the target as it moves inside the field of view of the camera under control.

In the smart networks of cameras for surveillance applications, most of the time, one camera is tracking an object (by estimating its position, orientation, and velocities); while the neighboring camera is patrolling. Before the object leaves the field of view of the tracking camera to the field of view of the neighboring camera, the tracking camera sends an alert to the neighboring camera to inform its neighboring camera that the object is approaching. Then, the

tracking camera starts sending observation measurements of the leaving object to the neighboring camera via communication links. Subsequently, the neighboring camera starts estimating the 3D position, orientation, and velocities of the object; and consequently, it knows when and where the object enters to its field of view ahead of time. Knowing the time and place where the object enters to the field of view of the neighboring camera is important for this camera because it can not move arbitrary fast. As a result, lack of this knowledge may result in tracking loss when the object enters to its field of view.

As discussed in Section 3.3, camera platform is based on FPGA and DSP on-board processors that have slow development cycles and limited computing flexibility. As a result, tracking algorithms must be developed in the presence of computation constraint.

Here, it is assumed that the target is moving in a 2D space. In order to describe the dynamic behavior of the target, the following unicycle model is used which is borrowed from [3].

$$\begin{aligned}
 x(t+1) &= x(t) + \cos(\phi(t))v(t)T \\
 y(t+1) &= y(t) + \sin(\phi(t))v(t)T \\
 \phi(t+1) &= \phi(t) + \omega(t)T \\
 v(t+1) &= v(t) + a(t)T \\
 \omega(t+1) &= \omega(t) + \alpha(t)T.
 \end{aligned} \tag{10}$$

Here,  $x \in \mathfrak{R}$  and  $y \in \mathfrak{R}$  are the target coordinates in the world reference system,  $\phi \in [0, 2\pi]$  the orientation,  $v \in \mathfrak{R}$  and  $\omega \in \mathfrak{R}$  the linear and angular velocity,  $a \in \mathfrak{R}$  and  $\alpha \in \mathfrak{R}$  the linear and angular acceleration, and  $T$  the sufficiently small sampling time (See Fig. 4). The accelerations are the inputs of the model which are unknown; but bounded:

$$|a| \leq a_{max} \quad |\alpha| \leq \alpha_{max}.$$

Since the accelerations are not known, they can be considered to be process noises. Moreover, the measurements are also affected by observation noises which are related to the camera accuracy. Let denote by  $\mathbf{x} = [x \ y \ \phi \ v \ \omega]^{tr}$  the state vector and by  $\mathbf{z} = [x \ y \ \phi]^{tr}$  the observation vector. Then, we can write the following dynamic model

$$\begin{aligned}
 \mathbf{x}(t+1) &= f(\mathbf{x}(t)) + \mathbf{w}(t) \\
 \mathbf{z}(t) &= h(\mathbf{x}(t)) + \mathbf{v}(t),
 \end{aligned} \tag{11}$$

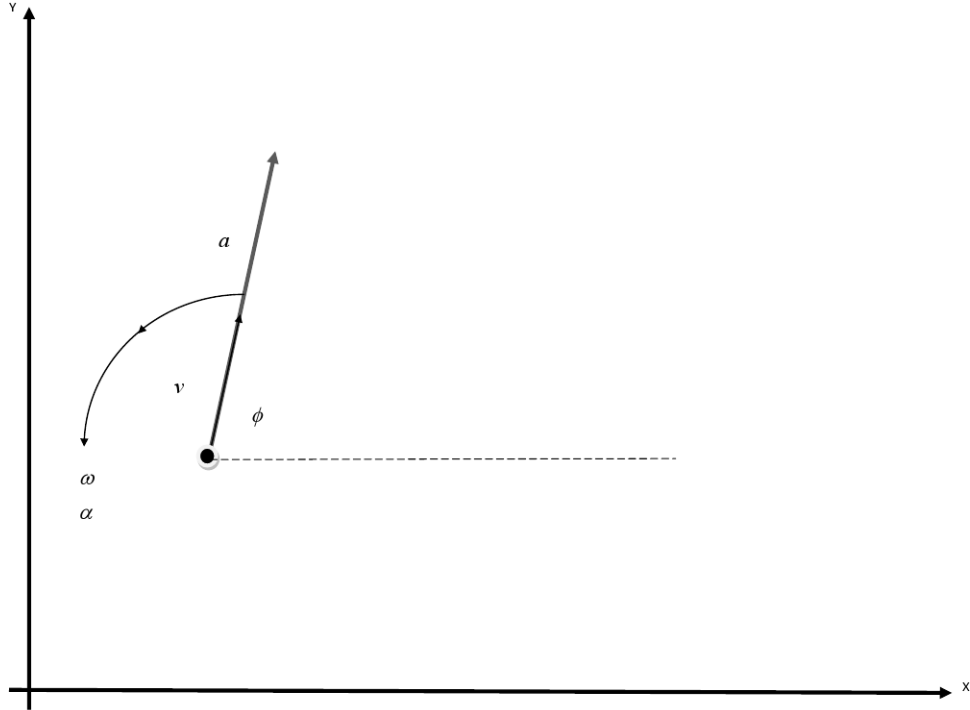


Figure 4: Unicycle model

where  $h(\mathbf{x}(t)) = [x \ y \ \phi]^{tr}$  and the process noise and the observation noise are respectively

$$\mathbf{w}(t) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ a(t)T \\ \alpha(t)T \end{pmatrix}, \quad \mathbf{v}(t) = \begin{pmatrix} v_x(t) \\ v_y(t) \\ v_\phi(t) \end{pmatrix},$$

while  $f$  is derived by (10). That is

$$f(\mathbf{x}(t)) = \begin{pmatrix} x(t) + \cos(\phi(t))v(t)T \\ y(t) + \sin(\phi(t))v(t)T \\ \phi(t) + \omega(t)T \\ v(t) \\ \omega(t) \end{pmatrix}.$$

Note that the initial state is assumed to have a bounded support, i.e.,  $\mathbf{x}(0) = (x_0 \ y_0 \ \phi_0 \ v_0 \ \omega_0)^{tr}$ , where  $|x_0| \leq u_{x_0}$ ,  $|y_0| \leq u_{y_0}$ ,  $|\phi_0| \leq u_{\phi_0}$ ,  $|v_0| \leq u_{v_0}$ ,  $|\omega_0| \leq u_{\omega_0}$ . For the process noise we also have  $|a(t)T| \leq a_{max}T$ ,  $|\alpha(t)T| \leq \alpha_{max}T$ . Also, the measurement noise is related to the camera accuracy, i.e.,  $|v_x(t)| \leq u_x$ ,  $|v_y(t)| \leq u_y$ ,  $|v_\phi| \leq u_\phi$ .



Currently, our industrial partner (Videotech) is using a dedicated megabit

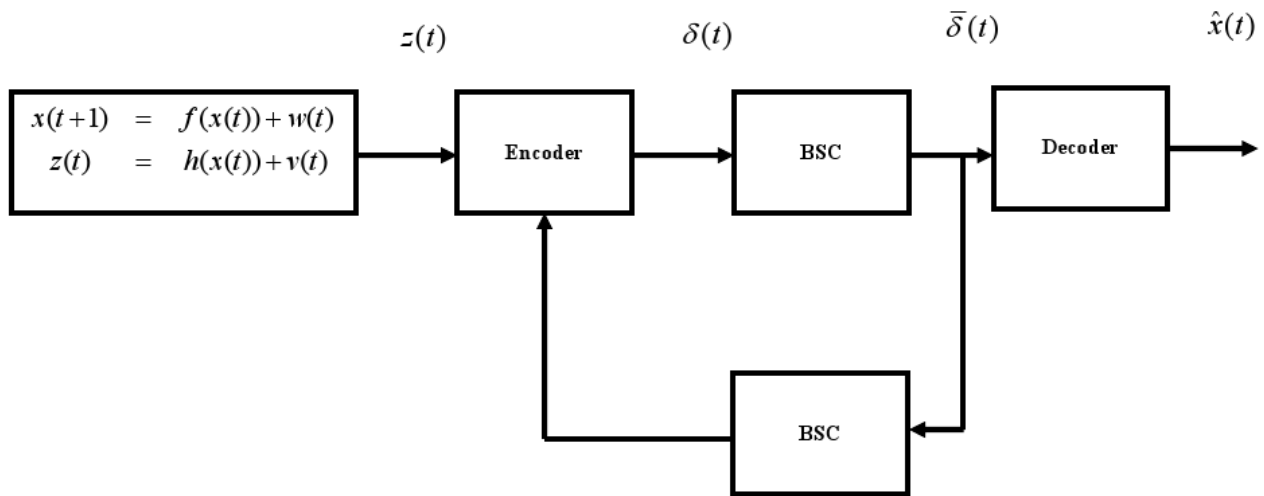


Figure 5: Communication system between two cameras

cable Ethernet network to interconnect cameras. As a result, cameras use power over Ethernet as the source of energy. Using this network, the communication occurs without latency and loss. However, in future, Videotech is interested to use wireless communication and solar panels/ batteries as the source of energy. Consequently, in this application, communication is subject to noise and limited bit rate. Fig. 5 illustrates the communication system for this application. Here, the communication channel is modeled by the binary symmetric channel, and  $\hat{\mathbf{x}}(t) = [\hat{x} \ \hat{y} \ \hat{\phi} \ \hat{v} \ \hat{\omega}]^{tr}$  denotes the state estimate. In this system, we can use feedback from the receiver to the transmitter. But, in general, this feedback is noisy. As discussed in Section 3.3, camera platform is based on FPGA and DSP on - board processors that have

slow development cycles and limited computing flexibility. As a result, in the communication system of Fig. 5, the coding algorithms must be developed in the presence of limited computation resource. The issue of limited computation resource can be addressed by developing recursive algorithms (which are not computationally expensive) with an increasing precision as time progress such that before the object leaves the field of view of the tracking camera, the neighboring camera knows the 2D position, orientation, and velocities of the object with high precision. This is the problem that we are interested to solve in the smart networks of surveillance cameras. That is, we are interested to develop coding algorithms (via designing proper encoder and decoder for the communication system of Fig. 5) which are recursive and have an increasing precision as time progress.

Note that in the communication system of Fig. 5, instead of sending observation measurements (i.e.,  $\mathbf{z}(t)$ ) to the neighboring camera, the tracking camera can send the estimated 2D position, orientation, and velocities of the object to the neighboring camera via the BSC, whenever a new update is available. But, reconstruction of the transmitted estimated 2D position, orientation, and velocities with high precision requires high bit rate and is computationally expensive. But, estimating the 2D position, orientation, and velocities using the transmitted observation measurements does not require a bit rate as high as the other case (as the dimension of  $\mathbf{z}(t)$  is less than  $\hat{\mathbf{x}}(t)$ ) and is not computationally expensive if recursive coding algorithms are used.

### 4.3 Existing EC2 results

Although there has been a significant progress in the development of integration framework for estimation/ communication and estimation/ computation co-designs in the last decade, there are few works that tried to develop an integration framework for control-estimation/ communication/ computation co-design [1, 5]. Como, Fagnani and Zampieri in [1] considered the problem of accurate, real time state estimation of the noiseless unstable linear system (12)

$$\begin{aligned} \mathbf{x}(t+1) &= A\mathbf{x}(t), \quad \mathbf{x}(0) \in \Gamma \subset \mathbb{R}^d \\ \mathbf{y}(t) &= \mathbf{x}(t), \end{aligned} \tag{12}$$

subject to a random valued initial state  $\mathbf{x}(0)$  with bounded support, over the binary erasure channel. In the binary erasure channel, the transmitted bit is either received correctly with probability  $1-p$  ( $0 \leq p \leq 1$ ); or erased (e.g, due to an unexpected collision) with probability  $p$ . In [1], anytime reconstruction technique [7] (i.e., the algorithm under which the reconstruction precision

increases with time) and two different linear coding schemes, as described below, are used to reconstruct the initial state  $\mathbf{x}(0)$ ; and subsequently,  $\mathbf{x}(t)$ , at the communication end.

For the simplicity of presentation of the algorithm developed in [1], let us assume  $d = 1$  (i.e.,  $\mathbf{x}(t) = x(t)$ ). Without loss of generality, Como, Fagnani and Zampieri in [1] assumed that  $x(0) \in [0, 1]$ . Consequently,  $x(0)$  has the following binary representation:

$$x(0) = \sum_{i=1}^{\infty} w_i 2^{-i}, \quad w_i \in \{0, 1\}.$$

In the first method, they used a random causal linear coding scheme which involves a lower triangular generating matrix  $M$ . In this scheme, the codeword  $z_t$  is the production of the following linear operation:

$$\begin{pmatrix} z_1 \\ z_2 \\ \cdot \\ \cdot \\ \cdot \\ z_t \\ \cdot \\ \cdot \\ \cdot \end{pmatrix} = M \begin{pmatrix} w_1 \\ w_2 \\ \cdot \\ \cdot \\ \cdot \\ w_l \\ \cdot \\ \cdot \\ \cdot \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \cdot & & & & & & & & \\ \cdot & & & & & & & & \\ \cdot & & & & & & & & \\ 1 & \dots & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ \cdot & & & & & & & & \\ \cdot & & & & & & & & \\ \cdot & & & & & & & & \end{pmatrix}, \quad (13)$$

where 0s and 1s on the lower triangular side of the matrix  $M$  are generated randomly; but the transmitter and the receiver know the components of this matrix a priori.

At the time instant  $t$ , the decoder receives  $\bar{z}_t$  which is the corrupted version of the codeword  $z_t$ . The decoder can use all the received codewords up to time  $t$ , i.e.,  $\bar{z}_1, \bar{z}_2, \dots, \bar{z}_t$  to reconstruct  $x(0)$ . For convenience, let denote the reconstructed version of  $x(0)$  at time  $t$  by  $\hat{x}(0|t)$ . In [1], the decoding law is linear and is described as follows:

For simplicity of presentation, let us assume that just  $\bar{z}_2$  contains erased bits; while other codewords are received correctly. Then, the decoder can use equation (14), as given below, to estimate  $x(0)$ .

$$\begin{pmatrix} z_1 \\ z_3 \\ \cdot \\ \cdot \\ \cdot \\ z_t \end{pmatrix} = \bar{M} \begin{pmatrix} w_1 \\ w_2 \\ \cdot \\ \cdot \\ \cdot \\ w_l \end{pmatrix}, \quad \text{where } \bar{M} \text{ is the matrix } M \text{ without the second row.} \quad (14)$$

For the decoder, the vector  $(z_1 \ z_3 \ \dots \ z_t)^{tr}$  and the matrix  $\bar{M}$  are known. Consequently, using the above equation, it estimates  $w_1, w_2, \dots, w_l$  as follows:

$$\begin{pmatrix} \hat{w}_1 \\ \hat{w}_2 \\ \cdot \\ \cdot \\ \cdot \\ \hat{w}_l \end{pmatrix} = (\bar{M}^{tr} \bar{M})^{-1} \bar{M}^{tr} \begin{pmatrix} z_1 \\ z_3 \\ \cdot \\ \cdot \\ \cdot \\ z_t \end{pmatrix},$$

and subsequently, it outputs  $\hat{x}(0|t) = \sum_{i=1}^l \hat{w}_i 2^{-i}$ .

In [1], it was shown that the maximum number of required operations at time  $t$  is  $\alpha t^3$ , for some  $\alpha > 0$ ; and the root mean square estimation error, i.e.  $\sqrt{E|x(0) - \hat{x}(0|t)|^2}$  is bounded above by  $Ct^{1/2}2^{-\beta t}$ , where  $C$  and  $\beta$  are positive constants which are dependent on the erasure probability  $0 \leq p < 1$  and the average transmission rate  $\mathcal{R}$ .

Now, let  $\hat{x}(t|t)$  denote the estimation of  $x(t)$  at time  $t$ . From the dynamic model (12), it follows that  $\hat{x}(t|t) = A^t \hat{x}(0|t)$ ; and subsequently, we have the following:

$$\sqrt{E|x(t) - \hat{x}(t|t)|^2} = |A|^t \sqrt{E|x(0) - \hat{x}(0|t)|^2} \leq |A|^t C \sqrt{t} 2^{-\beta t} = C \sqrt{t} 2^{t(\log |A| - \beta)}.$$

Therefore, for a given  $|A|$  and  $p$ , the average transmission rate  $\mathcal{R}$  can be chosen such that  $C \sqrt{t} 2^{t(\log |A| - \beta)} \rightarrow 0$ . Thus, using this technique, accurate, real time state estimation is guaranteed. However, the number of required operations grows up rapidly with time. Therefore, this scheme may not be computationally feasible for some applications.

The second method proposed in [1] is based on repetition coding scheme. In this scheme  $y_1 = w_1, \dots, y_t = w_1 w_2 \dots w_l$ . At the decoder, the estimated value for  $w_i$  at time  $t$ , i.e.,  $\hat{w}_i(t)$ , is chosen as follows:

$$\hat{w}_i(t) = w_i(s), \quad \text{if } \exists s \leq t : w_i(s) \neq e,$$

where  $e$  stands for the erasure symbol; otherwise,  $\hat{w}_i(t) = 0$ .

Here, the number of required operations at time  $t$  is  $\gamma t$ ,  $\gamma > 0$ ; but the root mean square estimation error, i.e.,  $\sqrt{E|x(0) - \hat{x}(0|t)|^2}$ , is bounded above by  $B2^{-\epsilon t^{1/2}}$ , for some  $B, \epsilon > 0$ . Using this technique, we may not have accurate, real time state estimation. In particular, for  $|A| > 1$ , the estimation error is going to be unbounded asymptotically due to slow decay rate of the estimation error in estimation of the initial state. But, the proposed technique is computationally feasible for many applications. Note that the repetition coding scheme, as described above, can be also used for tracking over the

binary symmetric channel.

Another work addressing the problem of control subject to communication and computation constraints is [5]. Matveev and Savkin in [5] developed a control/communication/computation co-design framework for stability of the partially observed system (15), as given below, subject to external disturbances over discrete memoryless channels (e.g., the binary symmetric channel, the binary erasure channel, etc).

$$\mathbf{x}(t+1) = A\mathbf{x}(t) + B\mathbf{u}(t) + \mathbf{w}(t), \quad \mathbf{y}(t) = C\mathbf{x}(t) + \mathbf{v}(t), \quad (15)$$

where  $\mathbf{x}(t) \in \mathfrak{R}^n$  is the state,  $\mathbf{u}(t) \in \mathfrak{R}^o$  the control,  $\mathbf{w}(t) \in \mathfrak{R}^n$  the exogenous disturbance,  $\mathbf{y}(t) \in \mathfrak{R}^l$  the measurement output, and  $\mathbf{v}(t) \in \mathfrak{R}^l$  the sensor noise. The initial state  $\mathbf{x}(0)$  and the noise  $\mathbf{w}(t)$ ,  $\mathbf{v}(t)$  are random vectors. This system, in general, is unstable and is defined on the complete probability space  $(\Omega, \mathcal{F}, P)$ . In [5], it is assumed that there is a discrete memoryless channel connecting the system to the remote controller; while the control signals are applied to the system directly. The objective in [5] is to find encoder, decoder, and controller for stability in probability. That is,  $P(|\mathbf{x}(t)| \leq b) \geq a, \forall t$ , for some  $b \geq 0$ , and  $a \in [0, 1]$ , where  $|\cdot|$  denotes the Euclidean norm on  $\mathfrak{R}^n$ . It was shown that a necessary and sufficient condition for stability of the system in probability using a finite memory decoder-controller (i.e., when decoder-controller consume uniformly bounded computational resource) is that

$$\mathcal{C}_{0F} > H(A),$$

where  $\mathcal{C}_{0F}$  is the Shannon zero error capacity with feedback [2] and  $H(A) = \sum_{\{i; |\lambda_i(A)| > 1\}} \log |\lambda_i(A)|$ , is the eigenvalue rate ( $\lambda_i(A)$ s are the eigenvalues of the system matrix  $A$ ).

The Shannon zero error capacity of many noisy channels (e.g., the binary symmetric channel, the binary erasure channel) is zero even in the presence of feedback channel. Therefore, one consequence of the above result is that we can not stabilize the unstable system (15) over many noisy channels using a finite memory decoder-controller that consume uniformly bounded computation resource. This result is true even in the favorable case when the sensor noise is zero (i.e.,  $v(t) = 0$ ), the plant noise and the initial state are distributed over known small balls, and no memory limitations are imposed on the coder.

Next, [5] addresses the following question: What is the minimum requirement to the computational load that makes the stabilization over a discrete memoryless channel possible? In [5], it was shown that by employing a

decoder-controller with bounded expected values of the consumed computational resources, the stability in probability is guaranteed provided:

$$\mathcal{C} > H(A),$$

where  $\mathcal{C}$  is the Shannon (regular) capacity [2].

Unlike the Shannon zero error capacity, the Shannon capacity is not zero for noisy channels (e.g., for the binary symmetric channel  $\mathcal{C} = p \log p + (1 - p) \log(1 - p)$  bits per channel use, where  $p$  is the cross over probability). Therefore, one consequence of the above result is that over noisy channels, the minimum requirement to the computational load for stability in probability is that the decoder-controller has bounded expected values of the consumed computational resources.

#### 4.4 Potential aggregation of the existing results and the proposed problems

As formulated in the previous sections, the problem that we are interested to solve is the problem of accurate, real time estimation of the states of systems (9) and (11) over the binary symmetric channel. The computational complexity is a central issue here. In a favorable technique, the number of required operations grows up linearly with time (i.e., the number of required operations =  $\nu t$ , for some  $\nu > 0$ ); while the coding algorithm is anytime [7], i.e., the algorithm under which the reconstruction precision increases with time.

From the discussions provided in the previous sections it follows that the available co-design frameworks [1, 5] may be applicable to the tracking problems of Fig. 3 and Fig. 5.

- The block diagram of Fig. 5 is similar to the problem considered in [5]. In the tracking problem of Fig. 5, the observation signal is transmitted via a noisy channel (the BSC) to a remote controller (high level control). The high level control updates the reference center and radius, which can be viewed as the control signals, and sends them to AUV without communication constraints. Because of this similarity, from [5] it follows that using the coding techniques, which consume uniformly computational resources, do not result in a circular formation with a center reaching to the source flow.

Moreover

- The second coding technique of [1] can be extended to address the problem of tracking over the BSC which is the communication channel considered in Fig. 5.

## References

- [1] G. Como, F. Fagnani, and S. Zampeiri. Anytime reliable transmission of real - valued information through digital noisy channels. *SIAM J. Control Optim.*, 48:3903–3924, 2010.
- [2] T.M Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley and Sons, 1991.
- [3] D. Sturzenegger J. Lygeros D. M. Raimondo, S. Gasparella and M. Morari. A tracking algorithm for ptz cameras. *2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems*, Annecy, France, 2010.
- [4] A. Seuret L. B. Arranz and C. Canudas de Wit. Translation control of a fleet circular formation of auvs under finite communication range. *The 48th IEEE Conference on Decision and Control*, pages 8345–8350, 2009.
- [5] A. S. Matveev and A. V. Savkin. The problem of stabilization of networked systems under computational power constraints. *European Journal of Control*, pages 449–467, 2009.
- [6] B. J. Moore and C. Canudas de Wit. Source seeking via collaborative measurements by a circular formation of agents. *American Control Conference*, 2010.
- [7] A. Sahai and S. K. Mitter. The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication link - part i: scalar systems. *IEEE Transactions on Information Theory*, 52:3369–3395, 2006.