



Co-funded by the  
Erasmus+ Programme  
of the European Union

# MATHEMATICAL DESIGN OF WORLD IN PROGRAMMING

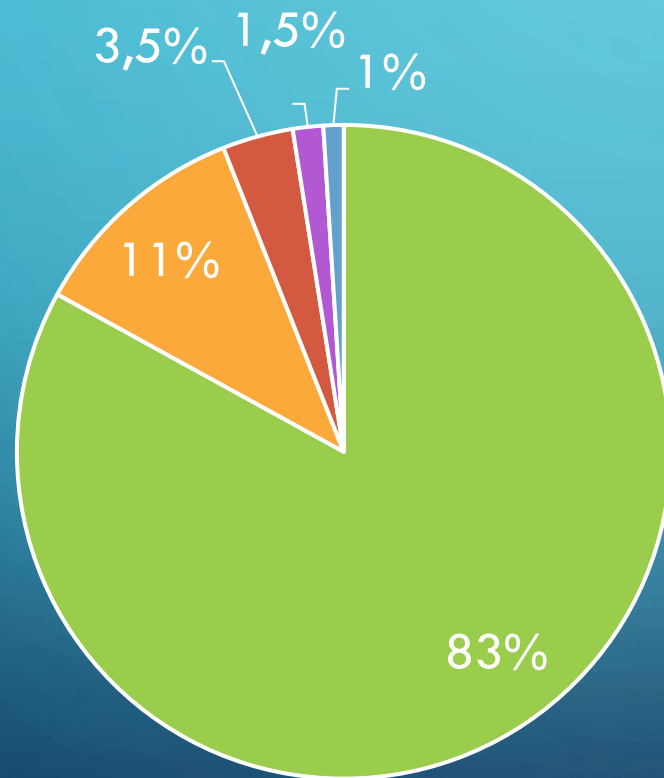


Maksymilian Wrzesień, PNA, Gdynia, Poland



# HOW DO WE RECEIVE INFORMATION?

## INFORMATION FROM THE SENSES



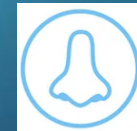
■ EYESIGHT



■ HEARING



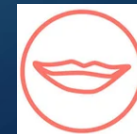
■ SMELL



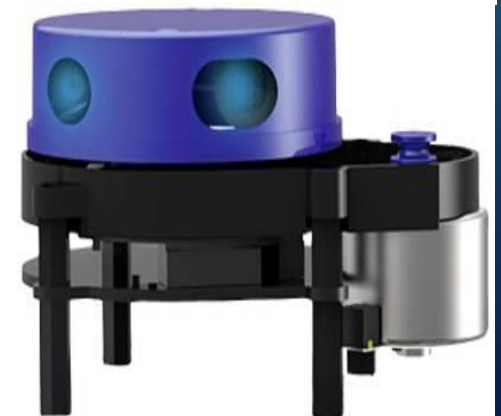
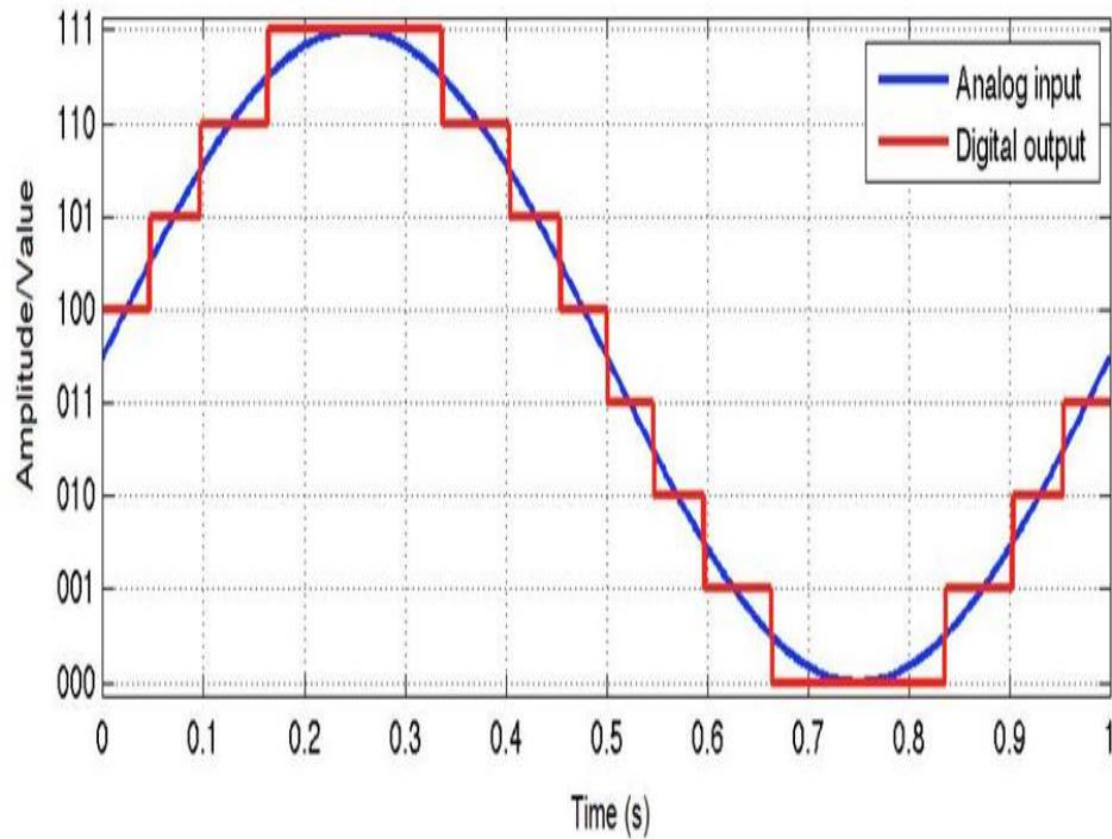
■ TOUCH



■ TASTE

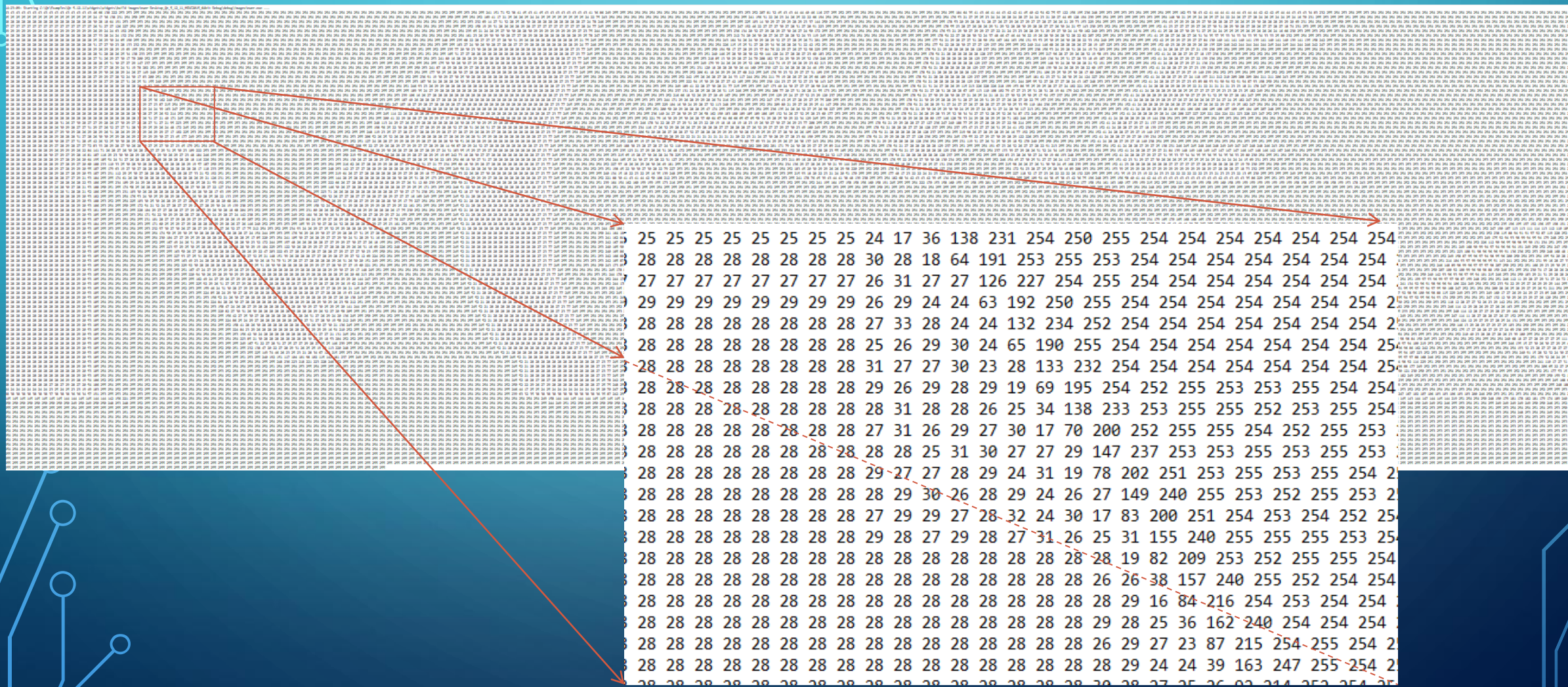


# HOW DOES COMPUTER RECEIVE INFORMATION?





# DATA STRUCTURES



25	25	25	25	25	25	25	24	17	36	138	231	254	250	255	254	254	254	254	254	254	254	254	254	
28	28	28	28	28	28	28	30	28	18	64	191	253	255	253	254	254	254	254	254	254	254	254	254	
27	27	27	27	27	27	27	26	31	27	27	126	227	254	255	254	254	254	254	254	254	254	254	254	
29	29	29	29	29	29	29	26	29	24	24	63	192	250	255	254	254	254	254	254	254	254	254	254	
28	28	28	28	28	28	28	27	33	28	24	132	234	252	254	254	254	254	254	254	254	254	254	254	
28	28	28	28	28	28	28	25	26	29	30	24	65	190	255	254	254	254	254	254	254	254	254	254	
28	28	28	28	28	28	28	31	27	27	30	23	28	133	232	254	254	254	254	254	254	254	254	254	
28	28	28	28	28	28	28	29	26	29	28	29	19	69	195	254	252	255	253	253	255	254	254	254	
28	28	28	28	28	28	28	28	31	28	28	26	25	34	138	233	253	255	255	252	253	255	254	254	
28	28	28	28	28	28	28	27	31	26	29	27	30	17	70	200	252	255	255	254	252	255	253	253	
28	28	28	28	28	28	28	28	28	25	31	30	27	27	29	147	237	253	253	255	253	255	253	253	
28	28	28	28	28	28	28	29	27	27	28	29	24	31	19	78	202	251	253	255	253	255	254	254	
28	28	28	28	28	28	28	28	29	30	26	28	29	24	26	27	149	240	255	253	252	255	253	253	
28	28	28	28	28	28	28	27	29	29	27	28	32	24	30	17	83	200	251	254	253	254	252	254	
28	28	28	28	28	28	28	28	29	28	27	29	28	27	31	26	25	31	155	240	255	255	255	253	254
28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	19	82	209	253	252	255	255	254
28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	26	26	38	157	240	255	252	254	254
28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	29	16	84	-216	254	253	254	254
28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	29	28	25	36	162	240	254	254	254
28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	26	29	27	23	87	215	254	255	254
28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	29	24	24	39	163	247	255	254	254
28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	28	27	25	26	92	214	252	254	254	

# DATA STRUCTURES






# NAUTICAL WORLD



NA

- **uSimPhinsINS:** An IxBlue INS simulator for navigation simulations.
- **uSimReplayPCAP:** A MOOS app that can replay a PCAP file over UDP.
- **pAcommsSim:** a GPU-accelerated multi-vehicle acomms simulator.
- **pSSView:** A viewer for the pSonarSim application.
- **pLidarSim:** GPU-accelerated lidar simulator.
- **pSonarSim:** GPU-accelerated sonar sim to predict time-domain sonar returns.
- **pGridViewer:** A MOOS viewer for Octree structures constructed from real or simulated point cloud data.
- **gen\_obstacles:** A command line tool for random generation of obstacle polygons to a file.
- **pMarinePID:** Simple PID controller for heading, speed and depth.
- **pMarineViewer:** Mission control GUI for monitoring, deploying and commanding vehicles.
- **pickpos:** A cmdline utility choosing simulation starting values.
- **uFldBeaconRangeSensor:** Simulates range information derived from pinging a buoy.
- **uFldCollisionDetect:** Monitors ship traffic collisions and reports near collisions.
- **uFldCollObDetect:** Monitors ship traffic obstacle collisions and reports near collisions.
- **uFldContactRangeSensor:** Simulates range measurements to other moving contacts.
- **uFldMessageHandler:** Handles and unwraps incoming messages from other vehicles.
- **uFldShoreBroker:** Facilitates shoreside share connections to vehicle communities.
- **uFldNodeComms:** Shoreside MOOS App for conditionally sending messages between vehicles.
- **uFldObstacleSim:** Simulation of obstacle (re)generation and lidar sensing of obstacles.
- **uFldScope:** Shoreside MOOS App for monitoring user-chosen fields across vehicles.
- **uFldNodeBroker:** Facilitates vehicle share connections to shoreside community.
- **uPokeDB:** A command line app for poking the MOOSDB.
- **uQueryDB:** A command line app, connect to MOOSDB, check condition, then exit.
- **uSimMarine:** A simple marine vehicle simulator.
- **uTimerScript:** An app for generating scripted pokes to the MOOSDB.

# ALGORITHM

```
Open ▾   
//-----  
// Procedure: Iterate()  
//           happens AppTick times per second  
  
bool Odometry::Iterate()  
{  
    m_total_distance+=sqrt(pow((m_current_x-m_previous_x),2)+pow  
((m_current_y-m_previous_y),2));  
    Notify("ODOMETRY_DIST",m_total_distance);  
    if(m_total_distance>50&& m_total_distance<150)  
    {  
        Notify("DESIRED_HEADING",180);  
    }  
    return(true);  
}
```



Terminal

```

Starting 0 0 | 0 0 | 0 | init_config
Present 0 0 | 0 0 | 0 | n/a

```

pMarineViewer (MIT Version 17.7)

File BackView GeoAttr Vehicles AppCasting MOOS-Scope Action

Node	AC	CW	RW	App	AC	CW	RW
alder	132	0	0	uSimMarine	97	0	0
				pNodeReporter	10	0	0
				uProcessWatch	5	0	0
				pMarineViewer	10	0	0
				pHelmIvP	10	0	0

```

uSimMarine alder
Datum: 43.8253, -70.3304

Start Pose      Current Pose (NAV)  Current Pose (G
Heading: 180    Heading: 180        (same)
Speed: 0        Speed: 0
Depth: 0        Depth: 0
Alt: 0          Alt: 0
(X,Y): 0,0     (X,Y): 0,0
Lat: 43.8253   Lat: 43.8253
Lon: -70.3304  Lon: -70.3304
Present Buoyancy rate: 0.025
Starting rate: 0.025
Source: initial_config

Ext Drift X Y Mag Ang Rot. Source(s)
Starting 0 0 0 0 0 init_config
Present 0 0 0 0 0 n/a

Velocity Information:
DESIRED_THRUST=0 ==> SPEED=0
Using Thrust Factor: true (20)
Positive Thrust Map: n/a
Negative Thrust Map: n/a
Max Accereration: 0

```



VName:  X(m):  Lat:  Spd:  Dep(m):  Time:

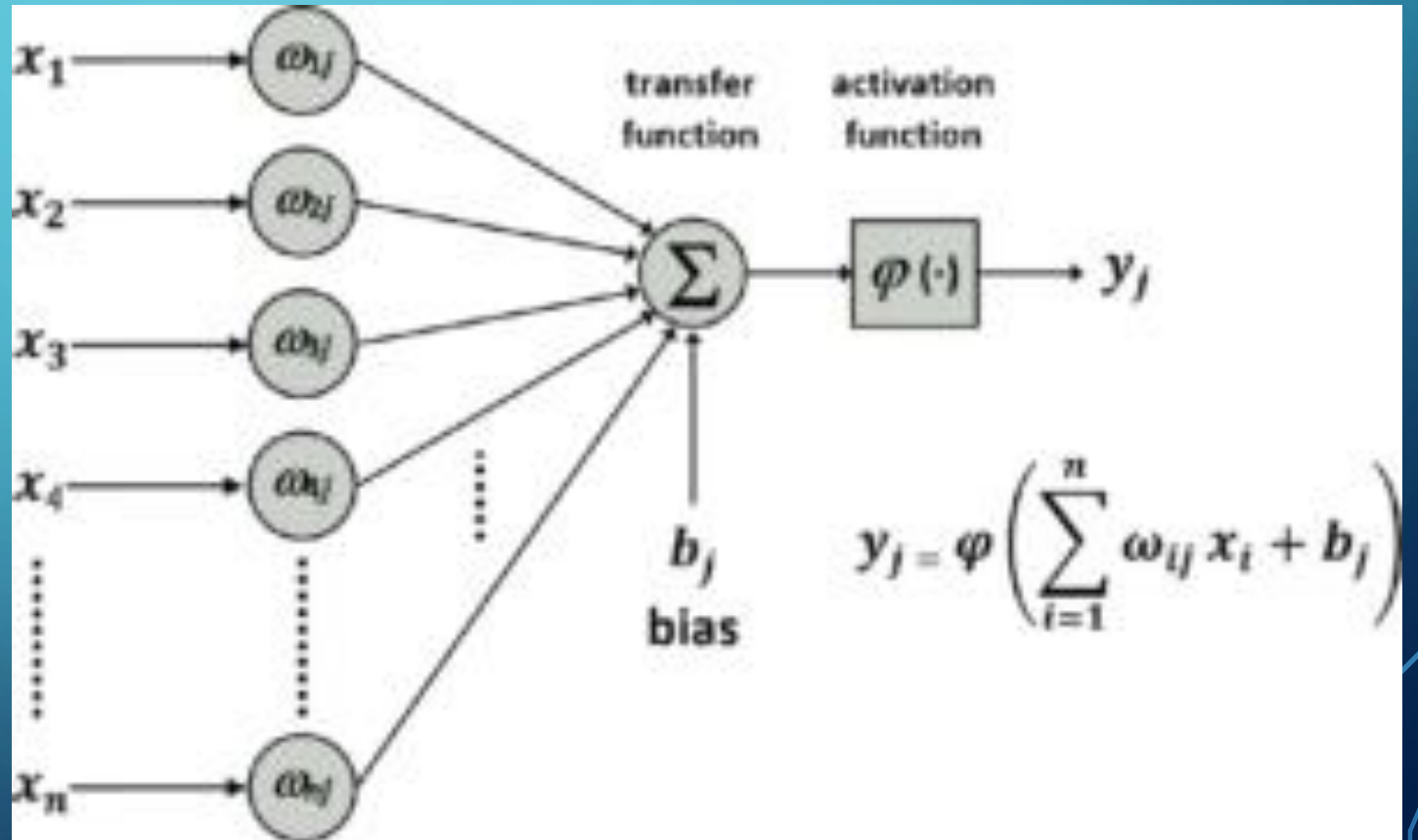
VType:  Y(m):  Lon:  Hdg:  Age(s):  Warp:

/variable:  Val:



# USAGE OF DATAS

- Optimization
- Recognition of sound
- Computer vision
- Artificial Inteligence





THANK YOU FOR YOUR ATTENTION

