Signals, Instruments, and Systems – W8

Introduction to Embedded Systems – Communication and Mobility
Outline

• Communication in embedded systems
  • Wired
  • Wireless
  • Power

• Mobility in embedded systems
  • Motivation, autonomy, and robots
  • Control architectures for mobile robots:
    • A simple taxonomy
    • The obstacle avoidance example
Motivation from Week 1 Lecture

Highlighted blocks are those mainly leveraging the content of this lecture.
Wired Communication
Where?

• Within embedded systems (from sensor to microcontroller, from microcontroller to microcontroller, etc.)
• From an embedded system to another
• From an embedded system to a PC
• …
Communication Model

Transmitter → channel → Receiver
Communication Model

Transmitter

Noise

channel

Receiver

Modulation
Coding
(Compression)

Distortion
Filtering
Frequency shift
...

Demodulation
Decoding
(Decompression)
A Seminal Example: The RS-232 (serial port)

- Hardware:
  - 3 wires: TxD, RxD, Ground

Transceiver = Transmitter + Receiver
RS-232 (serial port)

- Signal: between RxD/TxD and Ground
RS-232 Modulation

![Diagram of RS-232 Modulation](image)
RS-232 Demodulation

Start
+15V

 LSB
 b0  b1  b2  b3  b4  b5  b6  b7  MSB

 Stop
 Space

Idle
-15V

Idle
-3V

Time
RS-232 Delay

• Packet-based
  – 1 byte (i.e. 8 bits)/ packet
  – 8 data bits + 2 control bits (start/stop) = 10 bits

• Transmission speed
  – max. 115'200 bits/s (bps)

• Propagation speed:
  – approx. c (speed of light)
RS-232 Delay

• Transmission delay
  – 10 bits / 115'200 bps = 86.8 μs

• Signal propagation delay (2 m cable)
  – 2 m / c = 6.6712819 ns

• Processing delay:
  – ~ 1 us (modulation, demodulation, processing)

• Total: ~ 90 μs = 0.09 ms
Wireless Communication
Communication Model

Transmitter

Noise

channel = ElectroMagnetic waves in air

Reflections
Fading
Interference
Other EM sources
...

Receiver
Communication Model

Transmitter

channel = ElectroMagnetic waves in air

Noise

Receiver

Reflections

Fading

Interference

Other EM sources

Channel estimation

Advanced modulation types

Coding and error correction
Sharing the Medium

1 → 3 → 2
Sharing the Medium

• TDMA
  – Time-Division Multiple Access
  – “You shut up while I talk“
  – Time allocation
    • Fixed, synchronized
      e.g. mobile phones (GSM)
    • Dynamic (check if channel is free)
      e.g. Wireless LAN (802.11b/g/n)
Sharing the Medium

• FDMA
  – Frequency-Division MA
  – e.g. FM radio channels
  – Frequency regulation
• OFCOM (CH)

1 2 3

allocated by OFCOM

bandwidth

frequency
Bandwidth

- Can be defined by the OFCOM for multiple channels for a given purpose (in the overall spectrum)
- Can be defined for a single channel as follow:

\[ B = \text{bandwidth} \]
\[ f_0 = \text{carrier (channel) frequency} \]
\[ f_L = \text{low cut-off frequency} \]
\[ f_H = \text{high cut-off frequency} \]

-3dB = 50% power (spectral density)
i.e. \(10 \log(0.5) = -3\text{dB}\)

-3dB = 70% amplitude (spectral amplitude)
i.e. \(20 \log(0.7) = -3\text{dB}\)
Bandwidth

- FM station broadcasting at 106.4 MHz
  → actually occupies 106.3 MHz – 106.5 MHz
  → Bandwidth = 200 kHz

- Mobile phone (GSM, 2G): 200 kHz (around 900 MHz)
- WLAN/WiFi: 5 MHz (around 2.4 GHz)
- Analog TV station: 6 MHz (around 180 MHz)

What does the bandwidth depend on?

Bandwidth [Hz] ↑ → Data rate (Throughput) [bits/s] ↑
Bandwidth
Sharing the Medium

• CDMA (spread spectrum)
  – Code-Division MA
  – Using different transmission codes
  – e.g. GPS, WiFi, smart phones (3G/4G), Zigbee
  – Interesting properties
    • Wide channels (less fading)
    • Concurrent communication
  – More complex demodulation
Throughput (bits/s)

- TDMA, FDMA, CDMA can be combined
- Total throughput is shared
Shannon-Hartley Limit

• Hard theoretical limit on throughput
  – More bandwidth = higher throughput
  – More power (SNR) = higher throughput

\[ C = B \log_2 \left( 1 + \frac{S}{N} \right) \]

C: capacity (throughput)
B: bandwidth
S: signal power (W)
N: noise power (W)

Bit energy to noise-power spectral density \( \sim S/N \)
Power

• Increased power
  – higher throughput
  – higher range
  – mobile systems: shorter battery life
  – increased health risk (?)

• Regulation
  – CH: OFCOM
  – e.g. WLAN: 100 mW
Power

• Unit: W (Watt)
  – Often written in dBm (decibels to 1 mW)

\[ P_{dBm} = 10 \log_{10}(P_{mW}) \]

• Gain / loss: factors
  – Often written in dB (decibels)

\[ F_{dB} = 10 \log_{10}(F) \]
$P_{dBm}$ and Gain/Loss factors

$$P_{dBm} = 10 \log \left( \frac{P_W}{1\text{mW}} \right)$$

- $1\text{mW} \rightarrow 10 \log(1\text{mW}/1\text{mW}) \rightarrow 10 \log(1) = 10 \cdot 0 = 0 \text{ dBm}$
- $2\text{mW} \rightarrow 10 \log(2\text{mW}/1\text{mW}) \rightarrow 10 \log(2) \approx 10 \cdot 0.3 = 3 \text{ dBm}$
- $10\text{mW} \rightarrow 10 \log(10\text{mW}/1\text{mW}) \rightarrow 10 \log(10) = 10 \cdot 1 = 10 \text{ dBm}$
- $100\text{mW} \rightarrow 10 \log(100\text{mW}/1\text{mW}) \rightarrow 10 \log(100) = 10 \cdot 2 = 20 \text{ dBm}$

Factors in the chain become sums in a log form:
\[
\log(x \cdot y) = \log(x) + \log(y)
\]
## Link Budget

Typical WLAN link budget (100 m, dipole antennas):

<table>
<thead>
<tr>
<th>Component</th>
<th>Value</th>
<th>Margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>TX power</td>
<td>100 mW</td>
<td>20 dBm</td>
</tr>
<tr>
<td>TX losses</td>
<td>*0.5</td>
<td>-3 dB</td>
</tr>
<tr>
<td>TX antenna gain</td>
<td>*1.6</td>
<td>+2 dB</td>
</tr>
<tr>
<td>Free space path loss</td>
<td><em>1.0106</em>10^{-8}</td>
<td>-80 dB</td>
</tr>
<tr>
<td>RX antenna gain</td>
<td>*1.6</td>
<td>+2 dB</td>
</tr>
<tr>
<td>RX losses</td>
<td>*0.5</td>
<td>-3 dB</td>
</tr>
</tbody>
</table>

**RX power** 0.000000064 mW  **RX sensitivity** 0.0000000003 mW

**Margin** 200  **23 dB**
Free Space Path Loss (Friis Law)

- Signal power decay in air:
  \[ L = \left( \frac{4\pi df}{c} \right)^2 \]
  \[ L_{dB} = 20 \log_{10}(d) + 20 \log_{10}(f) - 147.56 \]

- Proportional to the square of the distance \( d \)
- Proportional to the square of the frequency \( f \)
  - high frequency = high loss
  - low frequency = low throughput
Ex.: DISAL Arduino Xbee kit vs. TinyNode

**DISAL Arduino Xbee kit**
- Microcontroller:  
  - ATMega 2560
- Transceiver:
  - Silicon Labs EM357 (part of the Xbee 802.15.4 module)
  - 2.4 GHz carrier
  - Throughput: up to 250 kbps
  - Range: up to 90 m

**TinyNode (Shockfish)**
- Microcontroller:  
  - TI MSP430
- Transceiver:
  - Semtech XE 1205
  - 868 and 915 MHz carriers
  - Throughput: up to 153 kbps
  - Range: up to 2 km
Mobility in Embedded Systems – Motivation, Autonomy, and Robots
Motivation

- Automation is progressing also in environmental and civil engineering
- Because of the size of the area in which mission must be accomplished (e.g., sensing, monitoring, acting) mobility is key
- Being mobile add complexity and cost at the node level but extend coverage of potentially expensive assets
- Mobile robots become progressively essential tools for automating environmental and civil engineering missions
Autonomy

• Different levels/degrees of autonomy
  – Energetic level
  – Sensory, motor, and computational level
  – Decisional level

• Needed degree of autonomy depends on task/environment in which the unit has to operate

• Environmental unpredictability is crucial: robot manipulator vs. mobile robot vs. sensor node
Simulated e-puck (Webots)
- sensor- and actuator-based
- noise, nonlinearities of S&A reproduced
- kinematic (e.g., speed, position) and dynamic (e.g., mass, forces, friction)

Real e-puck
Simulated and Real e-puck

• High-fidelity simulator Webots looks very powerful and realistic but:
  – Manufacturing heterogeneities not reproduced (e.g., all sensor of a certain type are the same, all the robots as well).
  – Noise distributions are typically uniform or Gaussian
  – Sensor field of view simplified (e.g., ray instead of cone)
  – Limitation in computational resources and internal electrical/computational architecture not reproduced
  – World physics approximated (e.g., geometry, communication channel, fluid dynamics) or not reproduced (e.g., chemical dispersion, thermal dissipation, etc.)
  – Real-time emulation very crudely approximated
  – …
Example: Real-Time Emulation

- **Sensors**
  - analog-digital conversion time
  - sampling rate

- **Environment**

- **Computation**
  - processing time

- **Actuators**
  - actuator delay
  - propagation time

**Delays are everywhere!**
Real-time Emulation

R = possible reality situation; S: possible simulation parametrization

R1

- Read and convert sensor data
- Normalize data
- Filter data
- Update actuators

100µs 60µs 135µs 120µs

R2

- Read and convert sensor data

30ms 200µs 120µs

S1

- Read and convert sensor data
- Normalize data
- Filter data
- Update actuators

32ms (sim step k-1) 32ms (sim step k) 32ms (sim step k+1)

S2

- Read and convert sensor data
- Normalize data
- Filter data
- Update actuators

64ms (sim step k-1) 64ms (sim step k) 64ms (sim step k+1)
A Simple Taxonomy for Control Architecture in Mobile Robotics
Perception-to-Action Loop for a Mobile Robot

Controller:
- Reactive vs. Deliberative
- Proximal vs. Distal

Perception • sensors

Computation

Environment

Action • actuators
Reactive vs. Deliberative Architectures

- **Reactive controller:**
  - 1 perception-to-action loop horizon
  - No planning, no history stored

- **Deliberative controller**
  - Multiple perception-to-action loop horizon
  - Planning and history exploitation

- **Reactive-deliberative boundary zone:**
  - Short history, short look-ahead horizon
  - A few state variables and little memory
Proximal vs. Distal Architectures

• Proximal:
  – close to sensor and actuators; no control hierarchy or layering
  – high flexibility in shaping the behavior by changing parameters and keeping the structure fixed
  – light architecture, fast execution time
  – works well when few resources available
  – difficult to engineer in a “human-guided” way
Proximal vs. Distal Architectures

- Distal:
  - farther from sensor and actuators; some control hierarchy or layering possible
  - less flexibility in shaping the behavior (behavioral module or basic behavior definition; flexibility only in the module “wiring” typically)
  - not always computationally light architecture
  - easier to engineer in a “human-guided” way because of the existence of modules (often hand coded)
Selected Reactive Architectures for Mobile Robots and their Application to Obstacle Avoidance
Overview

• Five “classical” examples of reactive control architecture for solving the same problem: obstacle avoidance.
• Two proximal: Braitenberg and Artificial Neural Network
• Three distal: Rule-based, Subsumption and Motor Schema, both behavior-based
Ex. 1: Braitenberg’s Vehicles

- Work on the **difference** (gradient) between sensors
- Originally **omni-directional** sensors; works also with **directional** sensors (sharper, potentially discontinuous differences at the sensory level -> more jerky movement)
- Originally: **light** sensors
- + excitation, - inhibition; **linear** controller (out = signed coefficient * in)
- Symmetry axis along main axis of the vehicle (----)
- See also lecture Week 7 and Lab 6
Ex. 1: Braitenberg’s Vehicles

With proximity sensors:
- $m = 8$
- local sensors -> must add a bias otherwise no stimulus

Linear summing & normalizing operator

$$O_i = Kf(x_i) = Kx_i$$

$K$: normalization constant

$$x_i = \sum_{j=1}^{m} w_{ij}I_j + I_0$$
Ex. 2: Artificial Neural Network

\[ O_i = K f(x_i) \]

\[ f(x) = \frac{2}{1 + e^{-x}} - 1 \]

\[ x_i = \sum_{j=1}^{m} w_{ij} I_j + I_0 \]

K: normalization constant
Ex. 3: Rule-Based

Rule 1:
if (proximity sensors on the left active) then
turn right

Rule 2:
if (proximity sensors on the right active) then
turn left

Rule 3:
if (no proximity sensors active) then
move forwards
Subsumption Architecture

- Rodney Brooks (1986)
- Precursors: Braitenberg (1984), Walter (1953)
- Behavioral modules (basic behaviors) represented by Augmented Finite State Machines
- Response encoding: predominantly discrete (rule based)
- Behavioral coordination method: competitive (priority-based arbitration via inhibition and suppression)
Augmented Finite State Machine

Behavioral Module

- Reset
- Suppressor
- Input lines
- Output lines
- Inhibitor: block the transmission
- Suppressor: block the transmission and replace the signal with the suppressing message
Ex. 4: Behavior-Based with Subsumption

Concrete implementation within basic behaviors:
Obstacle avoidance: Braitenberg without bias, rule-based, etc.
Wander: bias on motors (straightforward motion), random walk, etc.
Motor Schemas

• Ronald Arkin 1987, Georgia Tech
• Precursors: Arbib (1981), Khatib (1985)
• Parametrized behavioral libraries (schemas)
• Response encoding: continuous using potential field analog
• Behavioral coordination method: cooperative via vector summation and normalization
Ex. 5: Behavior-Based with Motor Schemas

Note: each motor schema generate an output vector that then gets summed up and normalized for controlling the actuators
Visualization of Vector Field for Ex. 5

Avoid-obstacle

Vector = [magnitude, direction]

\[ V_{\text{magnitude}} = \begin{cases} 
0 & \text{for} \quad d > S \\
\frac{S-d}{S-R} G & \text{for} \quad R < d \leq S \\
\infty & \text{for} \quad d \leq R 
\end{cases} \]

\( S = \) obstacle’s sphere of influence
\( R = \) radius of the obstacle
\( G = \) gain
\( D = \) distance robot to obstacle’s center

\( V_{\text{direction}} = \) radially along a line between robot and obst. center, directed away from the obstacle
Visualization of Vector Field for Ex. 5

Move-to-goal

Vector = [magnitude, direction]

$V_{\text{magnitude}} = \text{fixed gain value}$

$V_{\text{direction}} = \text{towards perceived goal}$
Visualization of Vector field for Ex. 5

Avoid-obstacle + move-to-goal

Linear superposition
(vectorial weighted sum)
Ex. 5: Issue with Motor Schemas

To avoid getting stuck in local minima of the vector field (typical problem of vector field approaches)

Alternative more complex approach: use harmonic potential functions (explicitly designed for not generating local minima)
Conclusion
Take Home Messages

• Communication plays a key role in embedded systems in general and in mobile robots in particular

• Some key concepts in communication systems
  – Bandwidth, real throughput, TDMA, FDMA, CDMA …
  – Transmitted/received power and corresponding losses

• The complexity of a controller depends on the degree of autonomy, mission to accomplish, and environmental conditions

• Controllers are characterized by parameters and a structure (architecture)

• A given overall behavior of the device can be obtained with different control architectures

• Controllers can be classified along two main axes: proximal vs. distal, reactive vs. deliberative
Additional Literature – Week 8

Pointers
• e-puck website: http://www.e-pucken.org/

Articles

Books