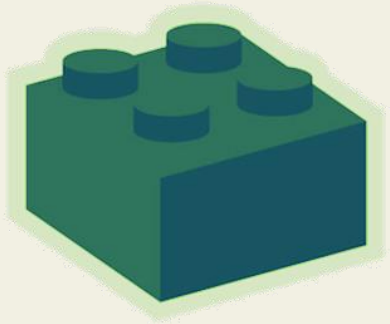
LEGO Mindstorms + C#

Horváth Ernő, Dr. Pozna Claudiu Radu



Tartalom

- Követelmény
- Előző évek
- Visual Studio, C#, XAML
- EV3 API
  - Motor
  - Szenzorok
- SLAM
- Kinematikai modell
- Mátrixok
- Kalman filter
- SVN

Elérhetőségek

Dr. Pozna Claudiu Radu
pozna@sze.hu

Horváth Ernő
http://www.sze.hu/~herno/
herno@sze.hu

Tanszéki honlap
http://it.sze.hu

RS1.SZE.HU/~HERNO/

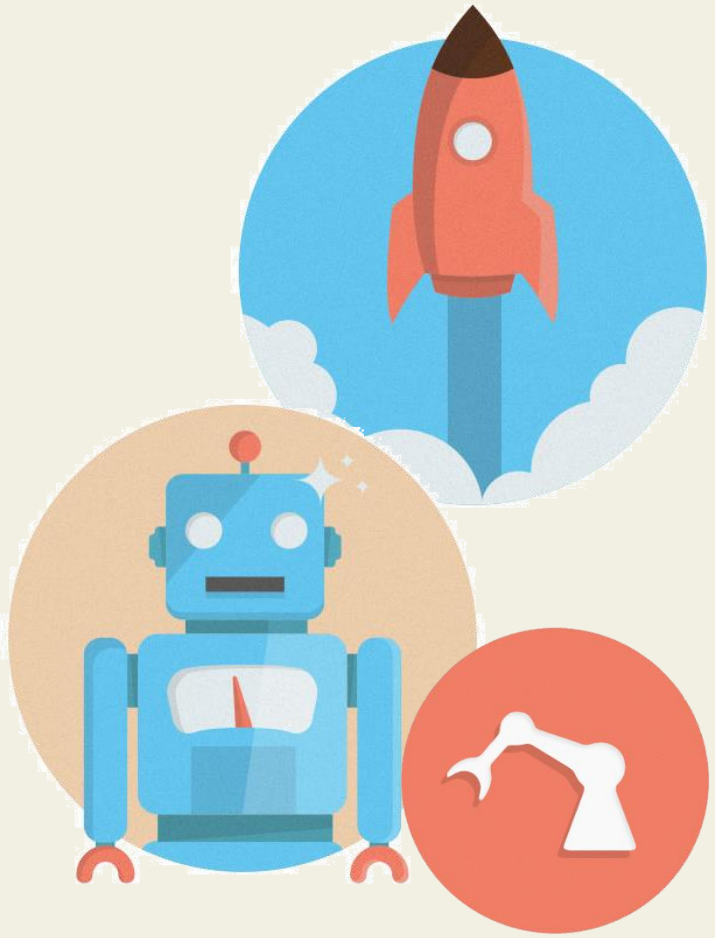
Követelmények

- Féléves jegy (szorgalmi időszakban szerezhető)
- ZH nincs
- Katalógus nincs (de az órán való aktív részvétel elengedhetetlen a jegy szerzéséhez)

Előző évek

- Khepera
- Neobotix projekt
- Tic-tac-toe

- Idén:
  - » Lego Mindstorms + SLAM

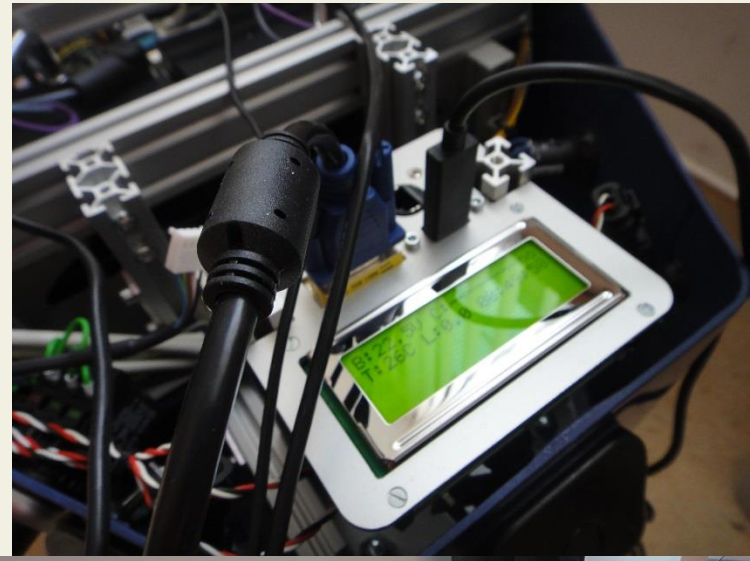


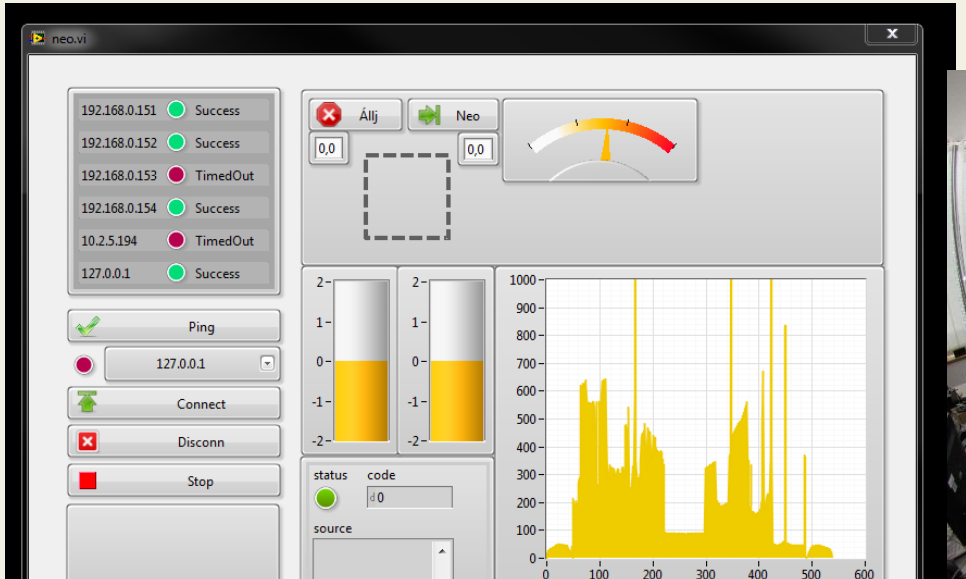
Projektmunka 1- **Khepera**

- Pályakövetése
- Elkerülési algoritmusok
- MATLAB

Projektmunka 2 - **Neobotix**

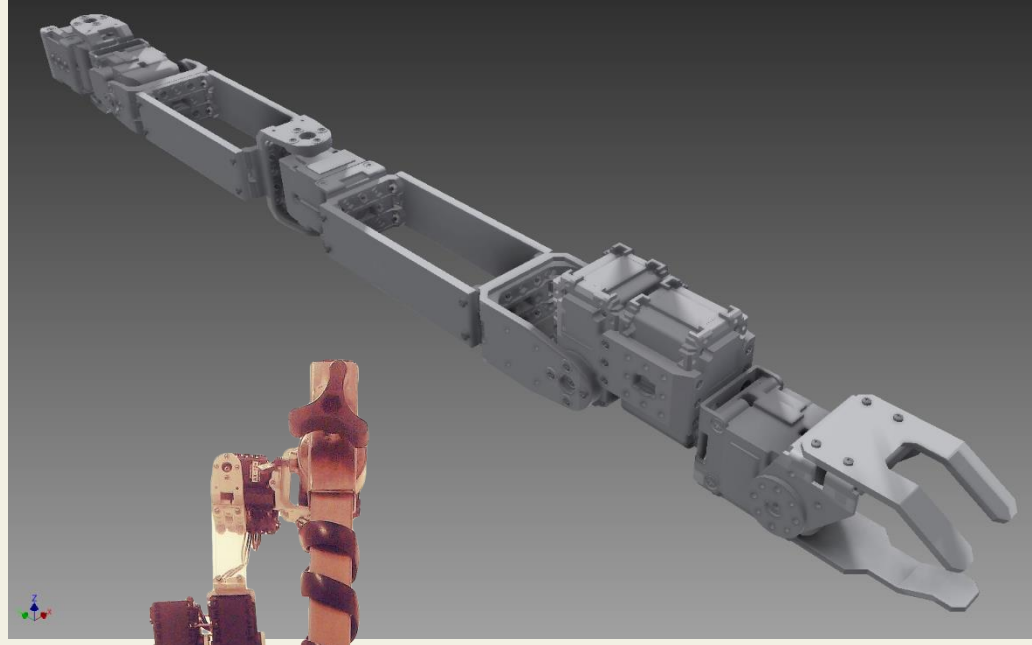
- ROS – Ubuntu Linux
- C#/Java
- LabVIEW
- HTML5

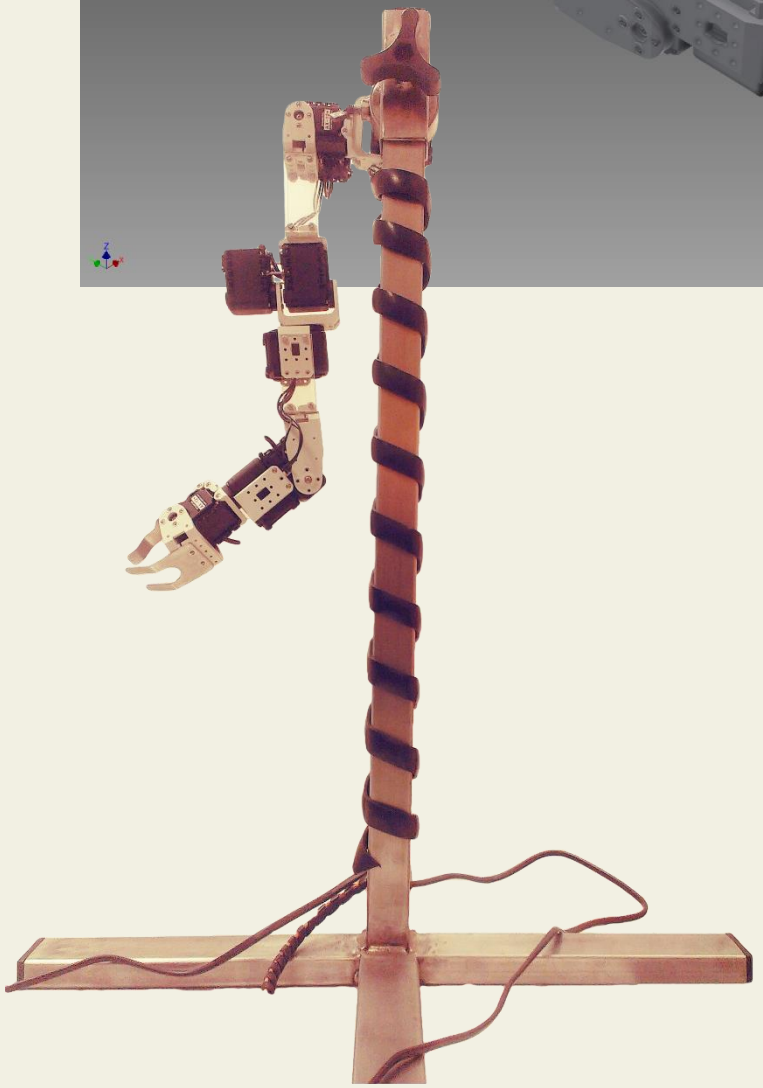




Projektmunka 3 - **Robotkar**

- Tic-tac-toe
- C#
- LabVIEW
- Képfeldolgozás
- Kinect
- http://www.sze.hu/~herno/#vids

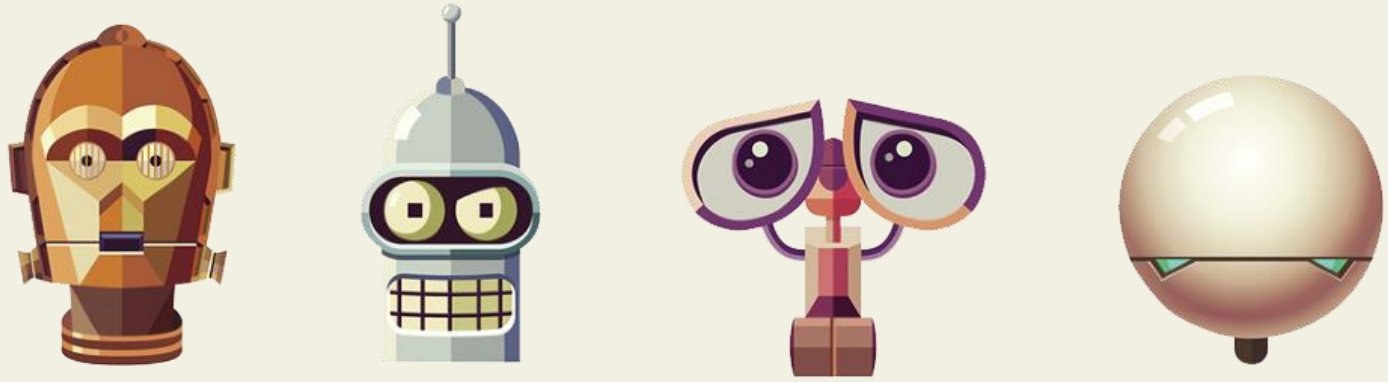




Robotika

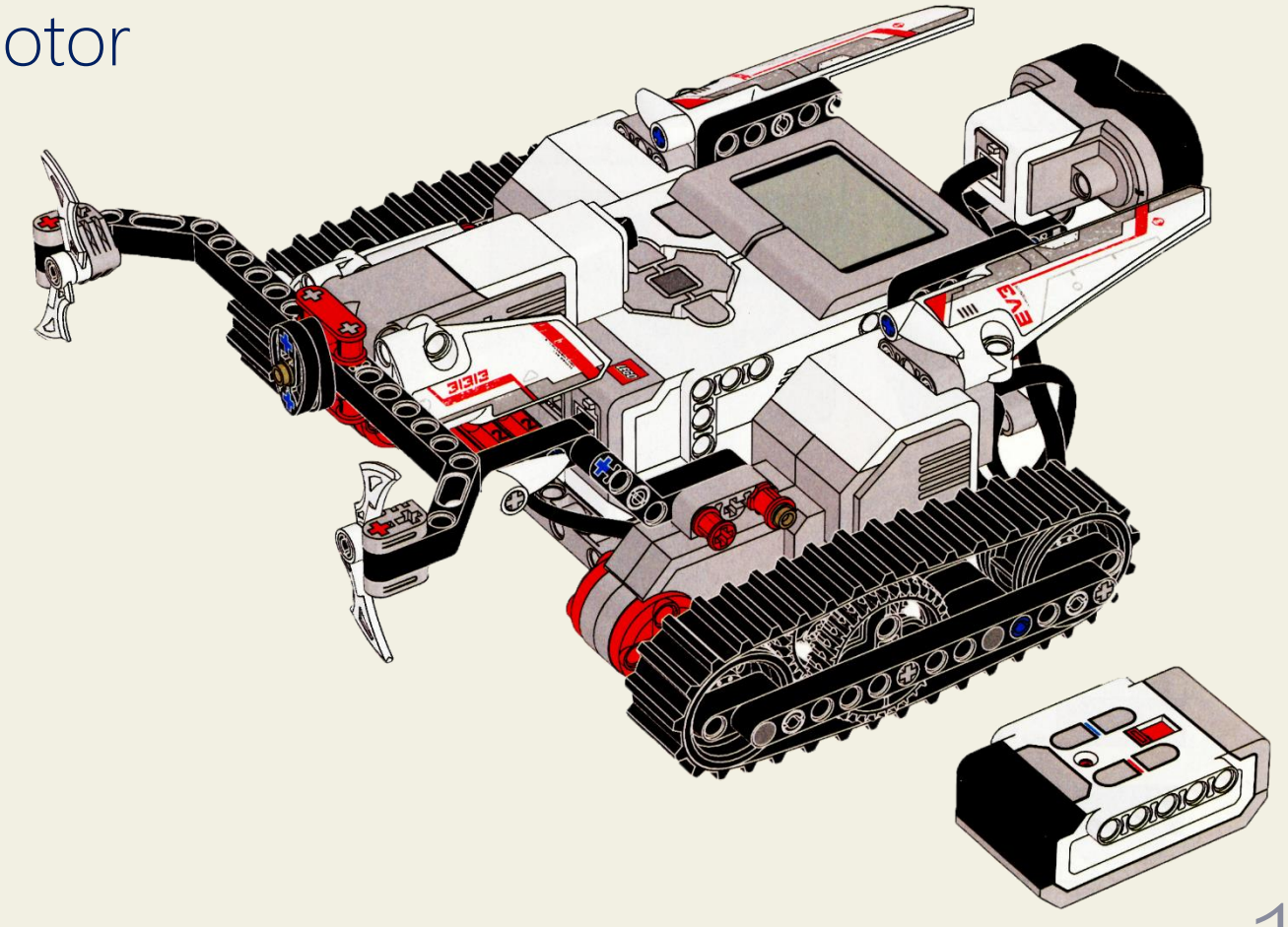
"Robotics is the branch of mechanical engineering, electrical engineering and computer science that deals with the design, construction, operation, and application of robots, as well as computer systems for their control, sensory feedback, and information processing. These technologies deal with automated machines that can take the place of humans in dangerous environments or manufacturing processes, or resemble humans in appearance, behavior, and/or cognition."

- Wikipedia

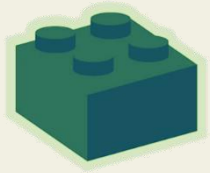


LEGO Mindstorms EV3

- Motorok és szenzorok
  - » Két nagy és egy közepes motor
  - » IR szenzor
  - » Szín szenzor
  - » Érintés szenzor

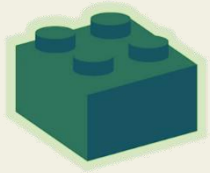


LEGO Mindstorms EV3



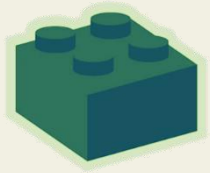
- 300 MHz, Texas Instruments, Sitara AM1808 (ARM9 core)
- 64 MB RAM, 16 MB Flash
- Bluetooth
- Motorok és szenzorok
  - » Két nagy és egy közepes motor
  - » IR szenzor (2 db)
  - » Ultrahang szenzor (1 db)
  - » Szín szenzor (2 db)
  - » Érintés szenzor (2 db)

C#



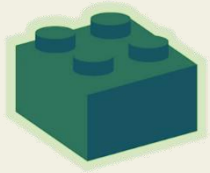
- C szintakszis az alapja
- Objektumorientált
- dotNET, Visual Studio

Visual Studio



- IDE
  - » Programkód szerkesztése
  - » Fordító (compiler), gépi kód modulokból programot létrehozó linker
  - » Debugger
  - » GUI designer
  - » Verziókezelés (pl. SVN)
  - » Stb.
- Programok, weboldalak, webszolgáltatások fejlesztésrére
- C, C++, C#, Visual Basic, stb.

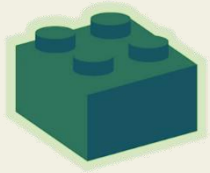
LEGO Mindstorms EV3 API (1.0.0)



- Kellően részletes dokumentáció
  - » http://www.sze.hu/~herno/NGB_IN039_1/2014Ev3/LegoMindstormsEv3Desktop/html/
  - » http://www.sze.hu/~herno/NGB_IN039_1/2014Ev3/LegoMindstormsEv3Desktop/chm/

```
using Lego.Ev3.Core;
using Lego.Ev3.Desktop;
```

Lego.Ev3.Core Namespace 1



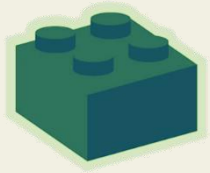
	Class	Description
	Brick	Main EV3 brick interface
	BrickButtons	Buttons on the face of the LEGO EV3 brick
	BrickChangedEventArgs	Arguments for PortsChanged event
	Command	Command or chain of commands to be written to the EV3 brick
	DirectCommand	Direct commands for the EV3 brick
	DummyCommunication	Dummy object for testing. Does not actually connect or communicate with EV3 brick.
	Port	An input or output port on the EV3 brick
	ReportReceivedEventArgs	Event arguments for the ReportReceived event.
	SystemCommand	Direct commands for the EV3 brick

Lego.Ev3.Core Namespace 2

	Interface	Description
	ICommunication	Interface for communicating with the EV3 brick

	Enumeration	Description
	BrickButton	Buttons on the face of the EV3 brick
	Color	UI colors
	ColorMode	EV3 Color Sensor mode
	ColorSensorColor	Values returned by the color sensor
	CommandType	The type of command being sent to the brick
	DeviceType	List of devices which can be recognized as input or output devices
	FontType	Font types for drawing text to the screen
	Format	Format for sensor data.
	GyroscopeMode	EV3 Gyroscope Sensor mode
	InfraredMode	EV3 Infrared Sensor mode
	InputPort	Ports which can receive input data

NuGet

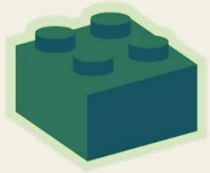


- Tools » Library Package Manager » Package Manager Console

```
PM> Install-Package Lego.Ev3
```

```
Successfully installed 'Lego.Ev3 1.0.0'.
Successfully added 'Lego.Ev3 1.0.0' to [solution].
```

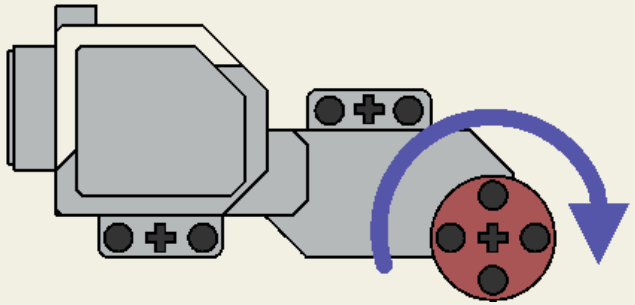
XAML



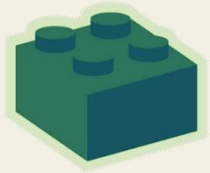
```
<Window x:Class="Ev3.Elso.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="LEGO példa" Height="350" Width="525" Loaded="Window_Loaded">
    <Grid>
        <StackPanel HorizontalAlignment="Left" Width="200">
            <Button Click="EloreClick">Előre</Button>
            <Button Click="HatraClick">Hátra</Button>
            <Button Click="BalClick">Balra</Button>
            <Button Click="JobbClick">Jobbra</Button>
            <Button Click="EmelClick">Emelés</Button>
            <Button Click="LerakClick">Lerakás</Button>
            <TextBlock x:Name="txtUltra" Text="Ultrahang" FontSize="30"></TextBlock>
        </StackPanel>
    </Grid>
</Window>
```

Motor

	EV3 nagy motor	EV3 közepes motor
Jeladó	1 fokonként	1 fokonként
Forgási sebesség	160 - 170 RPM (percenkénti fordulatszám) 2,5 - 2,83 Hz	240 - 250 RPM (percenkénti fordulatszám) 4- 4,1 Hz
Forgatónyomaték	0.21 Nm	0.08 Nm
Súly	76 g	36 g
Megjegyzés	Nagyobb tárgyak lassabb mozgatása	Kisebb tárgyak gyorsabb mozgatása



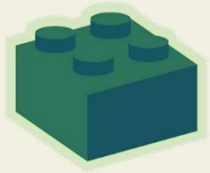
Motor



```
await _brick.DirectCommand.TurnMotorAtPowerForTimeAsync
                          (OutputPort.A, _fwd, _time, false);
```

- TurnMotorAtSpeedAsync
- TurnMotorAtSpeedForTimeAsync
- TurnMotorAtPowerAsync
- TurnMotorAtPowerForTimeAsync
- StepMotorAtSpeedAsync
- StepMotorAtPowerAsync
- StepMotorSyncAsync

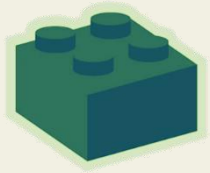
Motor



Egyszerre történő (kötegelt, batch) parancsküldés

```
_brick.BatchCommand.TurnMotorAtPowerForTime(OutputPort.B, _fwd,
_time, false);
_brick.BatchCommand.TurnMotorAtPowerForTime(OutputPort.C, _bck,
_time, false);
await _brick.BatchCommand.SendCommandAsync();
```

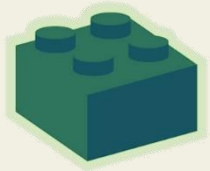
Motor



A motor pozíciójának lekérdezése.

```
float pos = brick.Ports[InputPort.A].SIValue;
```

Szenzorok



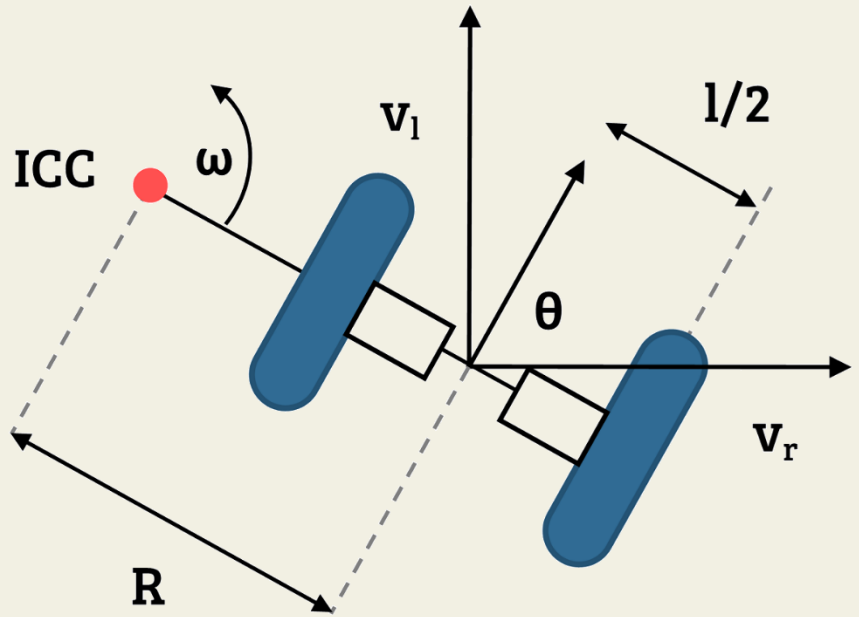
```
void _brick_BrickChanged(object sender, BrickChangedEventArgs e)
{
    System.Diagnostics.Debug.WriteLine("Esemény\n");
    txtUltra.Text = e.Ports[InputPort.Three].SIValue.ToString();
    txtMotor.Text = e.Ports[InputPort.A].SIValue.ToString();
    //throw new NotImplementedException();
}
```

SLAM

Simultaneous localization and mapping (SLAM) alatt olyan algoritmusokat értünk, amelyek egy mozgó eszközön futnak, egy terület bejárása során térképet hoznak létre (vagy frissítenek) miközben párhuzamosan, egyidejűleg nyomon is követik az adott eszköz pozícióját.

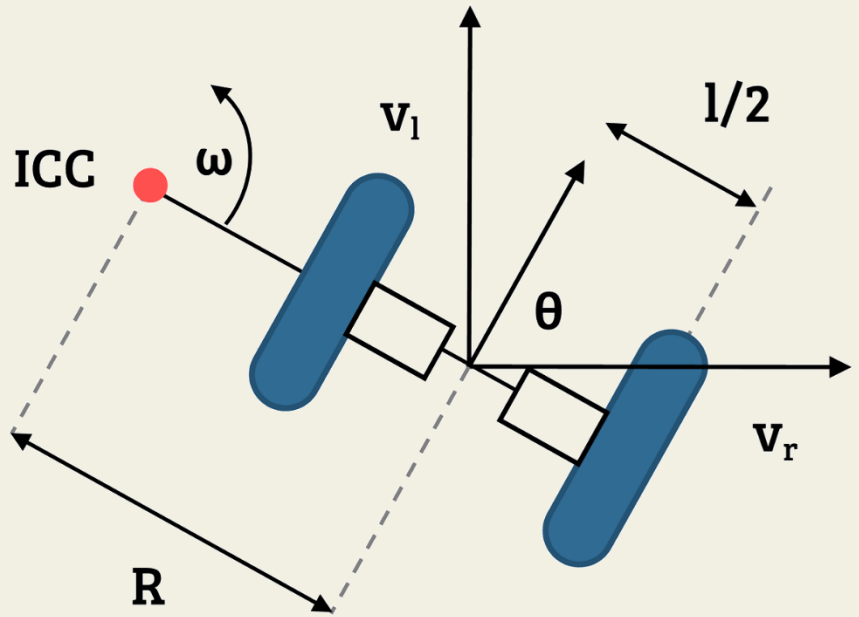
Kinematikai modell

- Differenciális, vagyis kerekenként eltérő meghajtás (differential drive) modell
- A bal kerék sebességet leíró V_l és a jobb kerék sebességet leíró V_r a következő egyenletekkel határozható meg:
- $V_l = \omega\left(R - \frac{l}{2}\right)$
- $V_r = \omega(R + \frac{l}{2})$



Kinematikai modell

- Esetünkben l jelenti a bal és a jobb kerék távolságát, R jelenti a pillanatnyi fordulási középpontot (ICC) és a kerekeket összekötő szakasz középpontja közti távolságot és ω pedig az elfordulás szögét. Ekkor az R és az ω meghatározható bármely időpontban:
- $R = \frac{l}{2} \frac{V_r + V_l}{V_r - V_l}$
- $\omega = \frac{V_r + V_l}{l}$



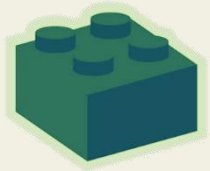
Kinematikai modell

A robot új pozíciója (x', y') és orientációja θ' a $t + \delta t$ időpillanatban, vagyis δt idő elteltével a következőképp számolható:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega\delta t) & -\sin(\omega\delta t) & 0 \\ \sin(\omega\delta t) & \cos(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} \begin{bmatrix} ICC_x \\ ICC_y \\ \omega\delta t \end{bmatrix}$$

A mozgásegyenletek sokat egyszerűsödnek, amennyiben csak a középpont körüli forgatást és az egyenes vonalú mozgást alkalmazzuk.

Kinematikai modell



Általánosan le tudjuk írni a Θ_t irányba és $V(t)$ sebességgel mozgó robotot adott t időpillanatban és pozícióban a következő egyenletekkel.

$$x(t) = \int_0^t V(t)\cos[\theta(t)]\,dt$$

$$y(t) = \int_0^t V(t)\sin[\theta(t)]\,dt$$

$$\Theta(t) = \int_0^t \omega(t)\,dt$$

Kinematikai modell

Az egyenletek differenciális meghajtású esetére a következőképp pontosíthatóak:

$$x(t) = \frac{1}{2}\int_0^t [v_r(t) + v_l(t)]\cos[\theta(t)]\,dt$$

$$y(t) = \frac{1}{2}\int_0^t [v_r(t) + v_l(t)]\sin[\theta(t)]\,dt$$

$$\Theta(t) = \frac{1}{l}\int_0^t [v_r(t) + v_l(t)]dt$$

Ebből adódik a kérdés miszerint hogyan tudjuk irányítani a robotot úgy, hogy elérjen egy adott pozíciót és irányt (x, y, θ)

Kinematikai modell

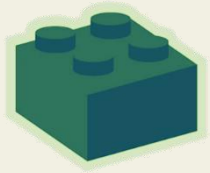
Ha mindkét kerék v sebességgel forog, tehát $v_r = v_l = v$

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x + v\cos(\theta)\delta t \\ y + v\sin(\theta)\delta t \\ \theta \end{bmatrix}$$

Ha pedig a középpont körül forgatjuk a robotot az x és y koordináta nem változik, csak a θ szög, mégpedig az δt idő, az l szélesség és a v sebesség alapján

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta + 2v\delta\, t/l \end{bmatrix}$$

Kinematikai modell



Pseudo kód

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x + v\cos(\theta)\delta t \\ y + v\sin(\theta)\delta t \\ \theta \end{bmatrix}$$

C# kód

```
x = (int)(x + v * Math.Cos(DegreeToRadian(theta)));
y = (int)(y + v * Math.Sin(DegreeToRadian(theta)));
```

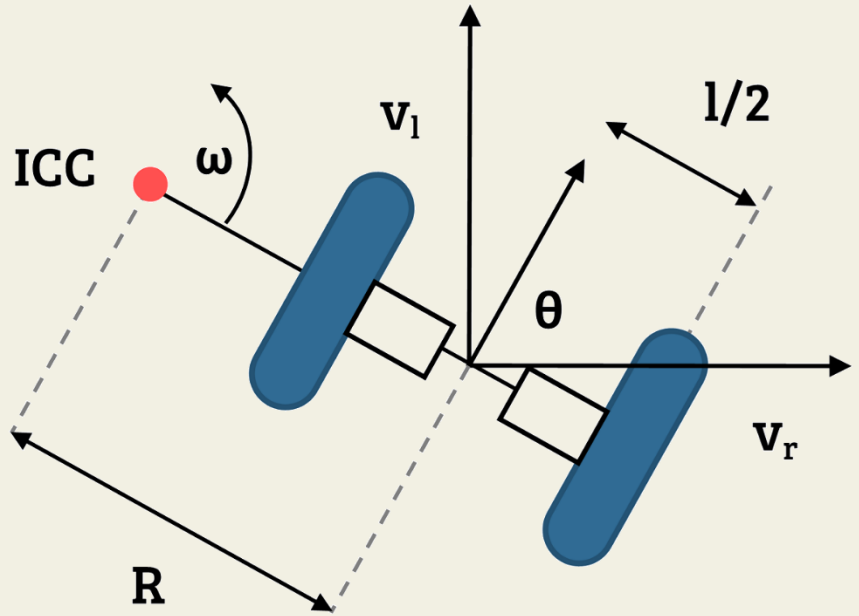
//egységnyi δt-vel számolva

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta + 2v\delta\, t/l \end{bmatrix}$$

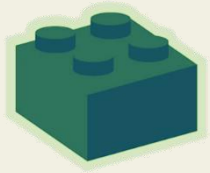
```
theta = theta + 2 * v / l
```

Milyen értékek szükségesek a modellhez?

Tulajdonság	*Érték*
A középponttól a kerék közepéig $\left(\frac{l}{2}\right)$	
Teljes szélesség: (D_{robot})	
Kerék teljes szélesség:	
Kerék futófelület szélessége:	
Kerék sugár: (r_{wheel})	
Két kerék távolsága (két kerék futófelületének a távolsága):	



Just in case! :)



Görög betűk

- δ - delta
- θ - theta
- ω - omega
- σ - sigma
- μ - mű

Mátrixszorzás

$$\begin{bmatrix} 2 & 3 & 4 \\ 1 & 0 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1000 \\ 100 \\ 10 \end{bmatrix} =$$

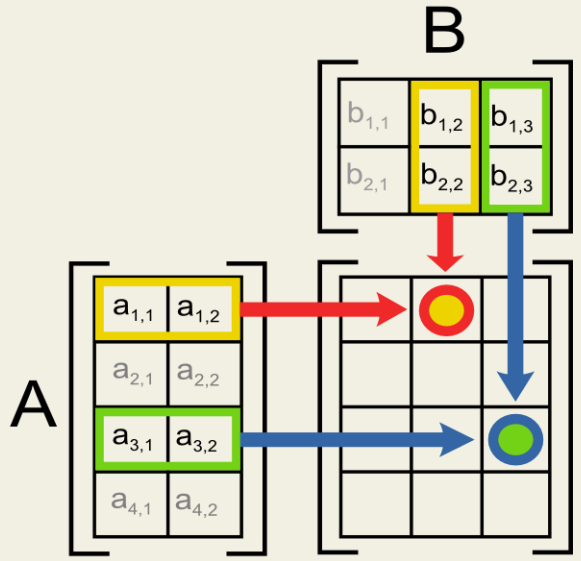
Just in case! :)

Görög betűk

- δ - delta
- θ - theta
- ω - omega
- σ - sigma
- μ - mű

Mátrixszorzás

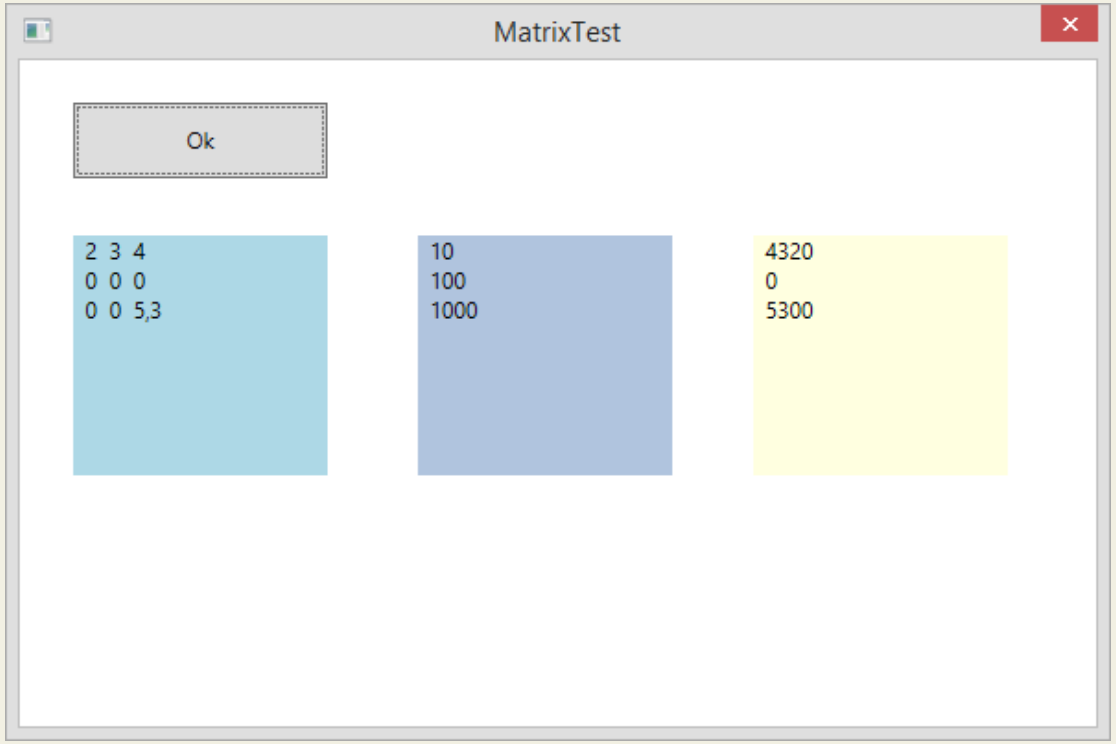
$$\begin{bmatrix} 2 & 3 & 4 \\ 1 & 0 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1000 \\ 100 \\ 10 \end{bmatrix} = \begin{bmatrix} 2340 \\ 1000 \\ 20 \end{bmatrix}$$

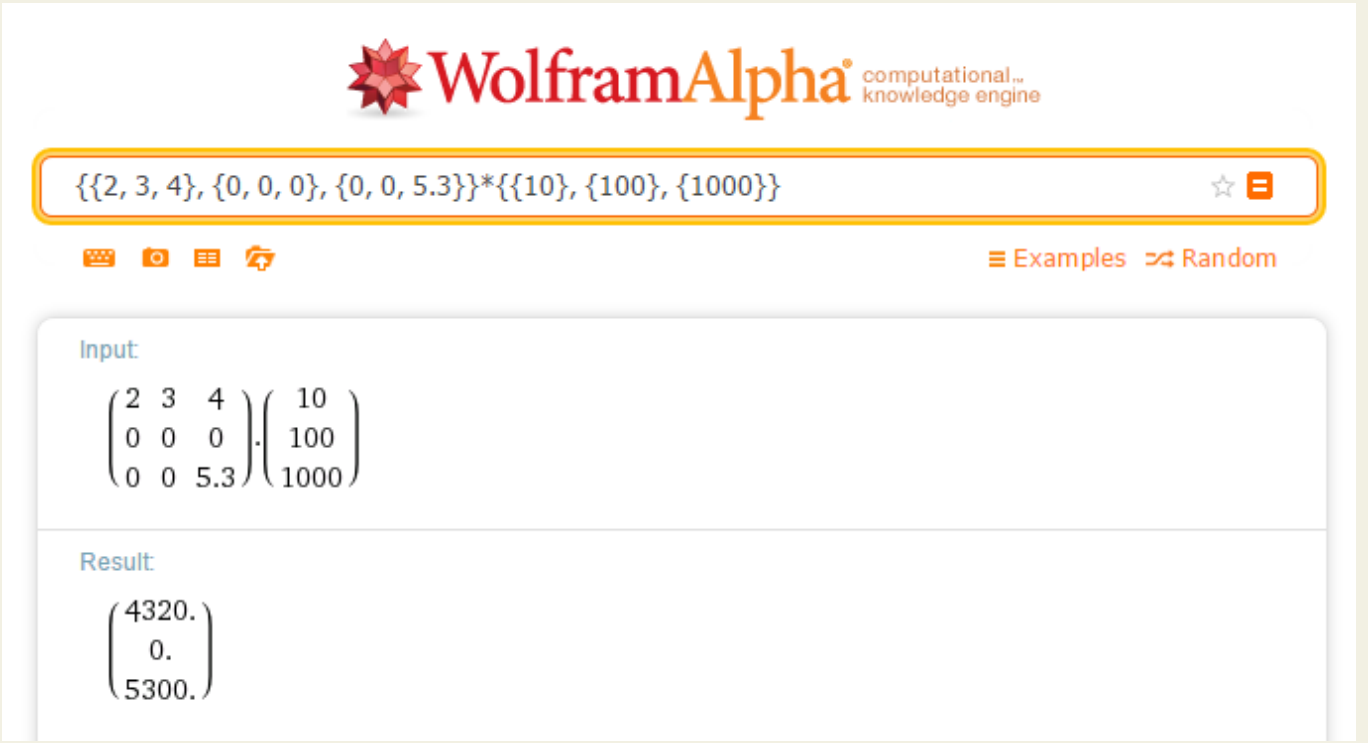


http://en.wikipedia.org/wiki/Matrix_(mathematics)

Mátrixok

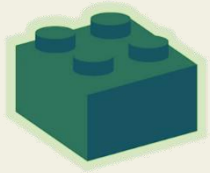
Mátrixszorzás bemutatása C# tesztprogramon keresztül





Ellenőrizzük: www.wolframalpha.com

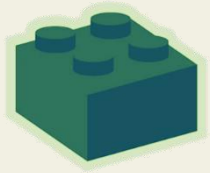
Mátrixok



A és B mátrix szorzása, eredménye C mátrix, csak akkor lehetséges, ha A mátrix oszlopainak száma megegyezik a B mátrix sorainak számával.

```
if (a.GetLength(1) == b.GetLength(0))
{
  c = new float[a.GetLength(0), b.GetLength(1)];
  for (int i = 0; i < c.GetLength(0); i++)
   {
    for (int j = 0; j < c.GetLength(1); j++)
     {
       c[i, j] = 0;
       for (int k = 0; k < a.GetLength(1); k++) // vagy k<b.GetLength(0)
           c[i, j] = c[i, j] + a[i, k] * b[k, j];
     }
   }
}
```

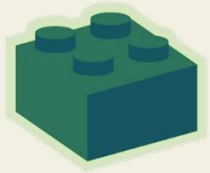
A RobotModel osztály



```
class RobotModel
  {
    private double _x;
    private double _y;
    private double _theta;
    private double _v;
    /// <summary>
    /// RobotModel létrehozása 3 paraméterrel
    /// </summary>
    /// <param name="_x">X pozíció</param>
    /// <param name="_y">Y pozíció</param>
    /// <param name="_theta">Orientáció</param>
    public RobotModel(double x, double y, double theta)
    {
        _x = x;
        _y = y;
        _theta = theta;
        _v = 10;
    }
```

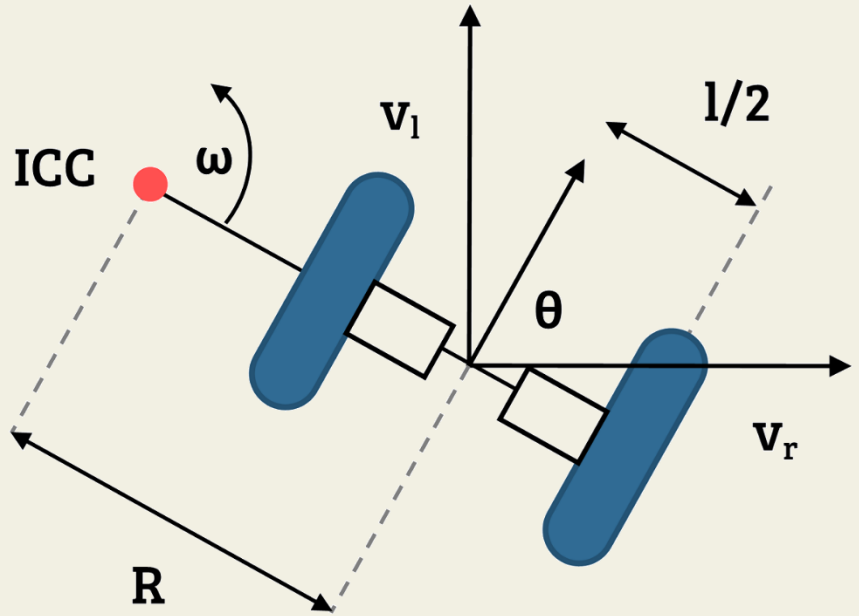
```
...
public void AddThetaDeg(float theta)
{
     _theta += theta;
     if (_theta < 0) _theta += 360;
     _theta = _theta % 360;
}

public float GetThetaDeg()
{
     return _theta;
}

public float GetThetaRad()
{
      return (float)(Math.PI / 180.0 * _theta);
}
...
```

A `RobotModel` osztály továbbfejlesztése

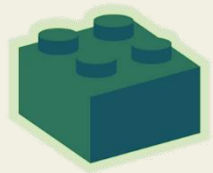


- `x, y,` θ
- `vr, vl`
- `ICCx, ICCy,` ω`, l`
- δ`t`
- `mátrixok`

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega\delta t) & -\sin(\omega\delta t) & 0 \\ \sin(\omega\delta t) & \cos(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} \begin{bmatrix} ICC_x \\ ICC_y \\ \omega\delta t \end{bmatrix}$$



SVN

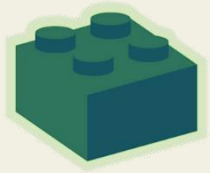


Angolul és szakszerű megfogalmazásokkal :) *(a kép egy előző évfolyam kommentjeit mutatja, nem a példát)*

Committed Changes

Rev	Scores	Commit log message	Date	Author
r32		Improved GUI and message handling.	Jan 19, 2014	beli
r31		Modified GameLogics module is now fully automatic (NOT TESTED YET)	Jan 18, 2014	beli
r30		organizing the merged projects	Jan 18, 2014	beli
r29		organizing the merged projects	Jan 18, 2014	beli
r28		organizing the merged projects	Jan 18, 2014	beli
r27		organizing the merged projects	Jan 18, 2014	beli
r26		Initial checkout of the merged project (WORKING!)	Jan 17, 2014	beli
r25		Rearrangements in the projects and dependencies, updated interfaces, some experimental code for storing and processing the game data.	Jan 13, 2014	beli
r24		[No log message]	Dec 24, 2013	hor
r23		[No log message]	Dec 24, 2013	hor
r22		[No log message]	Dec 24, 2013	hor
r21		[No log message]	Dec 24, 2013	hor
r20		pontosított mozgások 2 sor még hiány egyenlőre. és a mosolygó smiley :)	Dec 12, 2013	szil
r19		új pozíciók felvéve	Dec 3, 2013	szil
r18		[No log message]	Dec 2, 2013	hor
r17		test	Nov 26, 2013	dvi
r16		xml-ből olvas	Nov 26, 2013	szil
r15		új, javított, tényleges mozgások	Nov 22, 2013	szil
r14		Improved communication, test classes for debugging and easier implementation.	Nov 21, 2013	beli
r13		felveszi a bábút és leteszi egy megadott helyre	Nov 19, 2013	szil
r12		pozíció felvétel több mozgással	Nov 19, 2013	szil
r11		- updated events (as agreed during the team discussion) - some comments for the documentation	Nov 13, 2013	beli
r10		Null pozíció, megfogja a bábút és elviszi Null-ba.	Oct 29, 2013	szil
r9		Initial checkout of the game logic module.	Oct 24, 2013	beli
r8		Elmegy adott helyre, leereszkedik, de nem zárja össze a végén, majd nem mozgatja el. Az utolsó 2 mozgatást nem hajtja végre, amúgy király.	Oct 22, 2013	szil
r7		gyáááááááá :D	Oct 15, 2013	szil
r6		Szálmentes feladat, mozog, de nem áll be a célra, valahol megakad. Végtelen ciklus.	Oct 15, 2013	szil
r5		IsReady refactored, valószínűleg a szálakon hal el, random módon viselkedik.	Oct 15, 2013	szil
r4		mükődésre bírtuk, hiba: terminate jelenleg nem működik...	Oct 8, 2013	szil
r3		#01 upload project by Papp Szilárd	Oct 4, 2013	szil
r2		by Papp Szilárd	Oct 4, 2013	szil
r1		Initial directory structure.	Sep 19, 2013	---

Miért SLAM?

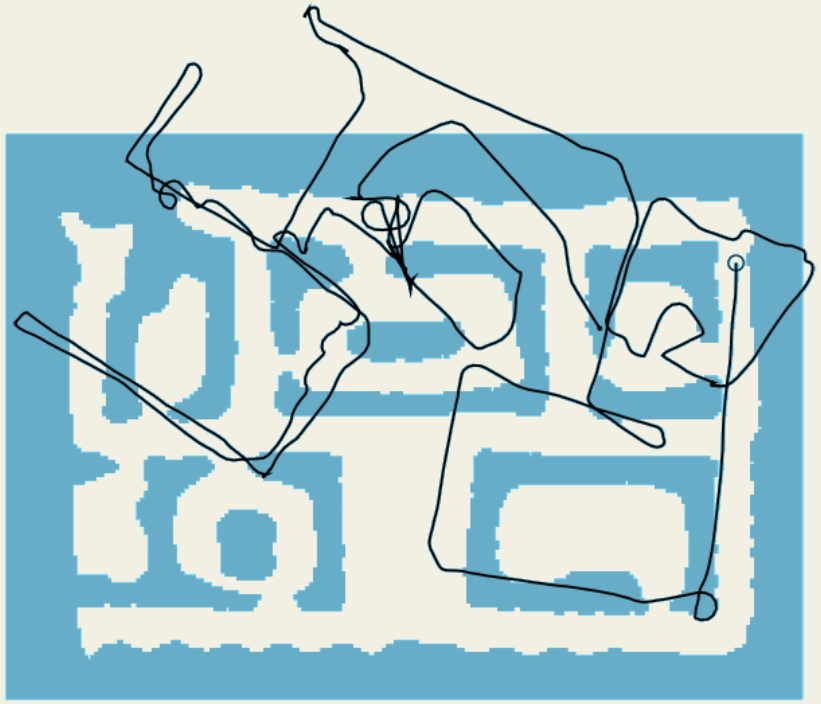


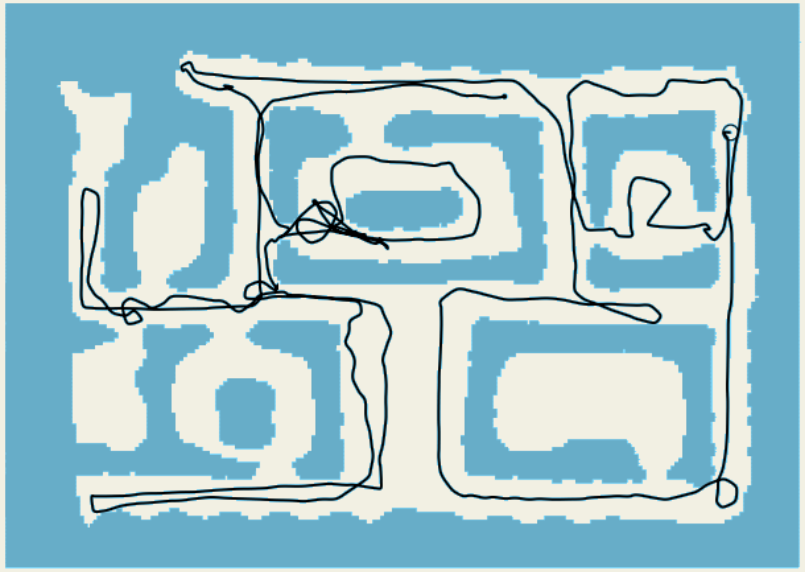
Problémák a pozícionálással / navigációval

- A robot mozgás modellje nem pontos
- A kerék csúszásából adódó hibák összeadódnak » kumulált hiba (nagyobb távolság » nagyobb hiba)
- Szenzor pontatlanságok

Szenzoros és becsült navigációs értékek

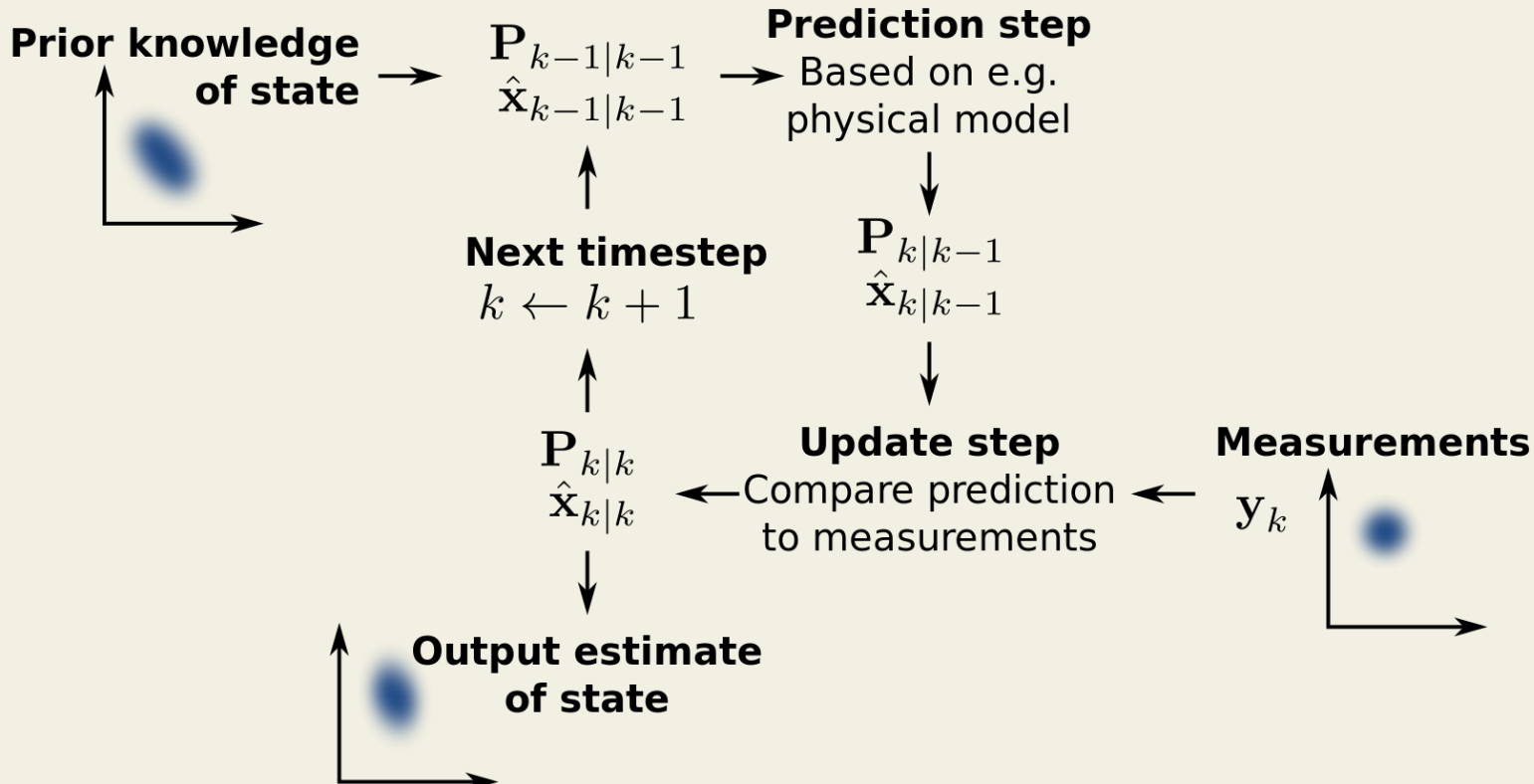
- A navigáció során számolt pozícióadatok (odometria) és a korrigált adatok



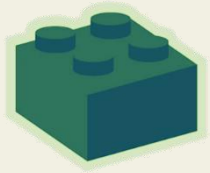


Kalman filter

- Kálmán Rudolf Emil (1930 -)
- Zajos bemenő adatok rekurzív mérésével egy pontosabb becslést ad a mérés tárgyának állapotáról.
  http://en.wikipedia.org/wiki/Kalman_filter



Bayes-tétel



Adott A és B események valószínűsége ($P(A)$ és $P(B)$), és a $P(B|A)$ feltételes valószínűség esetén fennáll a

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

egyenlőség.

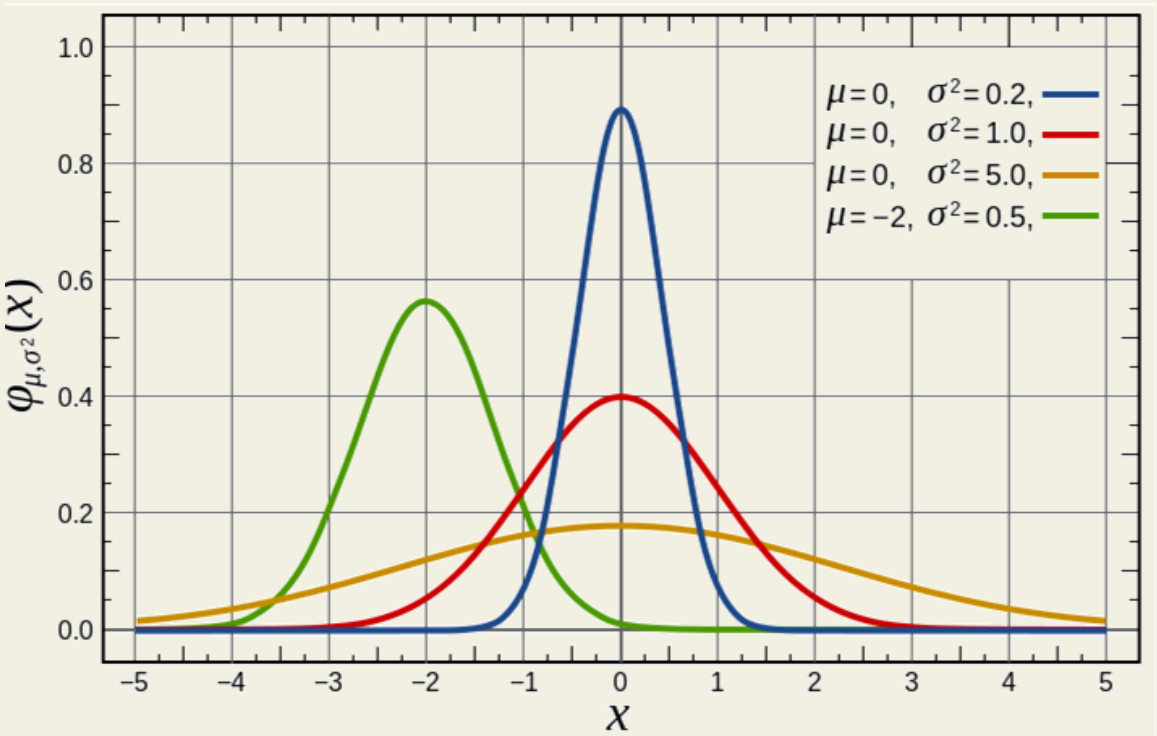
http://en.wikipedia.org/wiki/Bayes'_theorem

Normális eloszlás

Normal (Gaussian) distribution

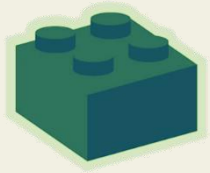
Sűrűségfüggvénye:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



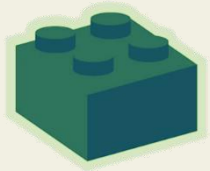
http://en.wikipedia.org/wiki/Normal_distribution

Kulcsszavak

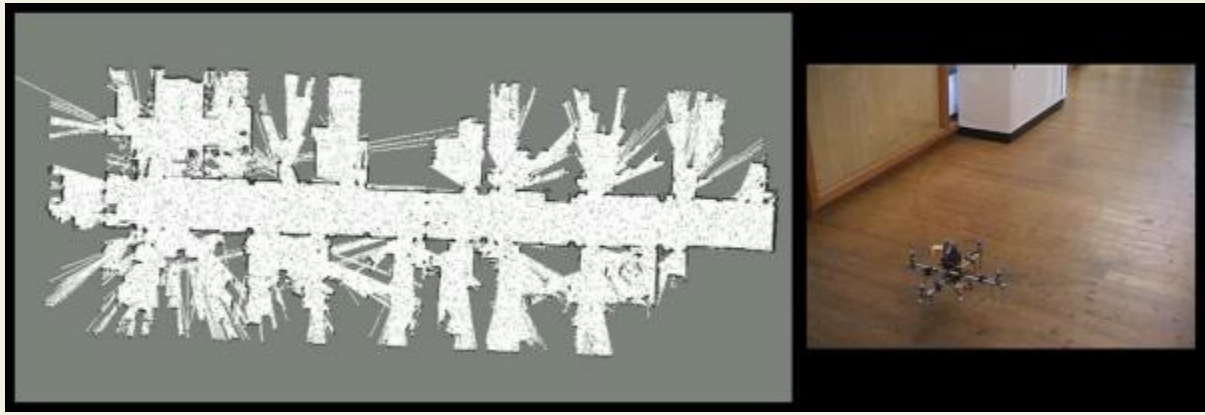


- Robot
  - » Segítség » Lego, Khepera, Neobotix
  - » Mechatronika
- Térkép
  - » Máté HTML5 canvas » C# canvas
    - Formátum
    - Zaj » Szűrés (lineáris regresszió)
- Táblázat, a robotok tulajdonságairól
    - I/O, programozás, szenzorok, tapasztalat

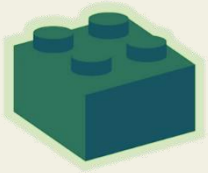
Videók



- Például:
  - » http://youtu.be/CYlb6ZcCSzU
  - » http://youtu.be/SeNLUW79
- Tutorial
  - » http://youtu.be/O5Zu19-tjY8
  - » http://youtu.be/h8gXn-E1ZFg



Felhasznált irodalom



- **A C programozási nyelv** - Az ANSI szerinti változat. **B. W.** Kernighan - **D. M. Ritchie**; Műszaki Könyvkiadó, 1995
- http://channel9.msdn.com/
- http://en.wikipedia.org/wiki/Robotics
- http://en.wikipedia.org/wiki/Simultaneous_localization_and_mapping
- http://www.probabilistic-robotics.org/