SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
FACULTY OF MECHANICAL ENGINEERING

Paper no. 1041
Virtual session

FloatShield: An Open Source Air Levitation Device for Control Engineering Education

Gergely Takács*, Peter Chmurčiak, Martin Gulan, Erik Mikuláš, Jakub Kulhánek, Gábor Penzinger, Marcel Vdoleček, Miloš Podbielančík, Martin Lučan, Peter Šálka and Dávid Šroba

IFAC
INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL

IFAC 2020 GERMANY

Institute of Automation, Measurement and Applied Informatics

Motivation: Commercial laboratory devices

- Teaching control engineering and mechatronics requires laboratory tools – “trainers” – for hands-on experience.
- Commercial tools are expensive, large, complicated and cannot be taken home by students.
- Many require closed-source software (e.g. MATLAB, LabView), and accessories (amplifiers, control PC, etc.)
- Implementation on microcontroller units (MCU) is under-represented



Motivation: Improvised laboratory devices

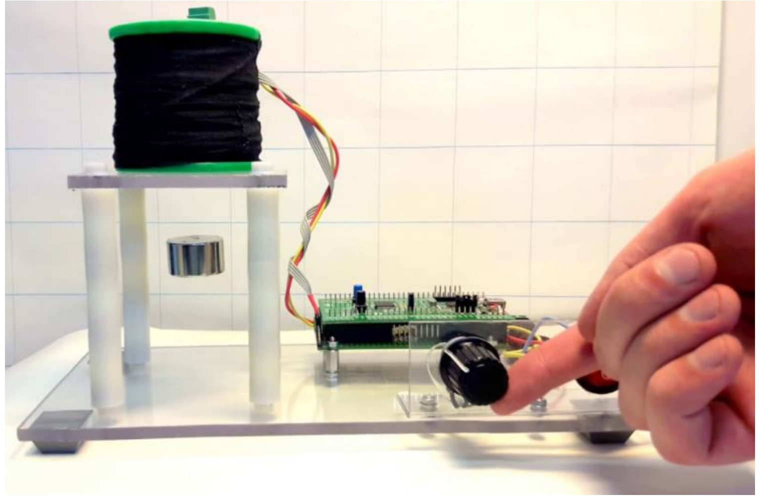
One of a kind improvised designs that are local to a laboratory or a small research team.

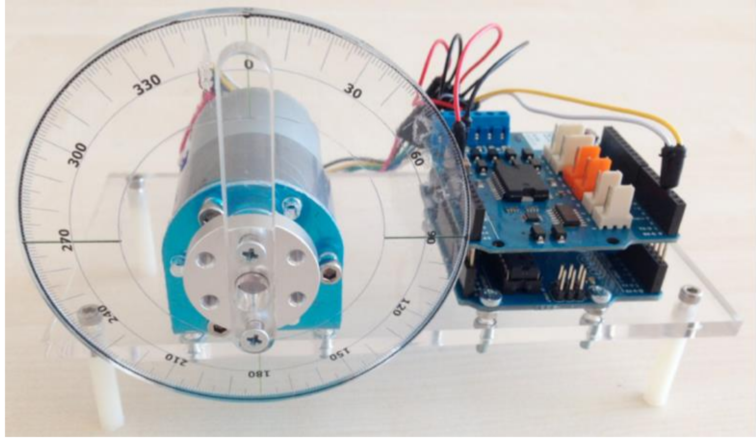
Pro:

- Cheap!

Contra:

- Fragile, sensitive setups
- Not very well documented
- Cannot create teaching materials across several universities as an open course





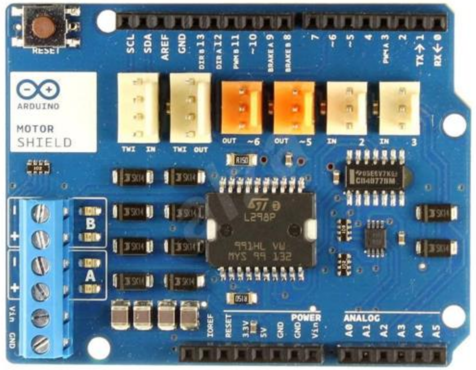
Motivation: Arduino, a universal platform to build on

- Cheap
- Open source
- Easy to buy
- Standardized
- Free integrated development environment (IDE)
- Great community and abundance of learning materials
- Easy hardware expansion through so- called Shields

ARDUINO
OPEN-SOURCE
COMMUNITY

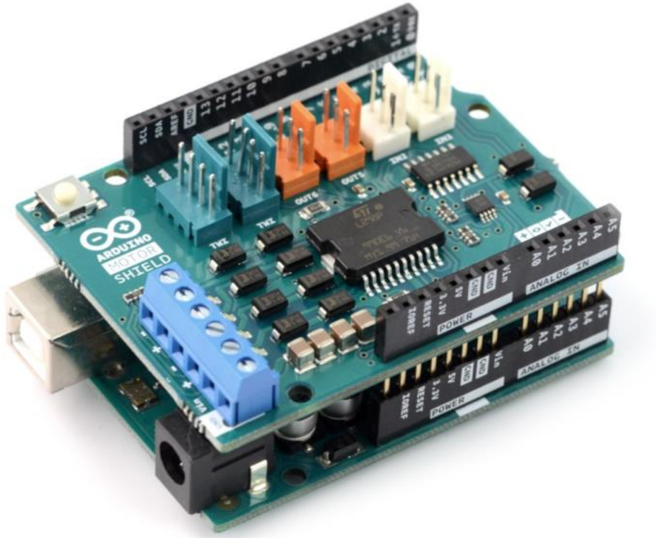
Arduino

+



Shield (Motor control)

=



Motivation: New tools for control engineering and mechatronics education

AutomationShield

Control Systems Engineering Education

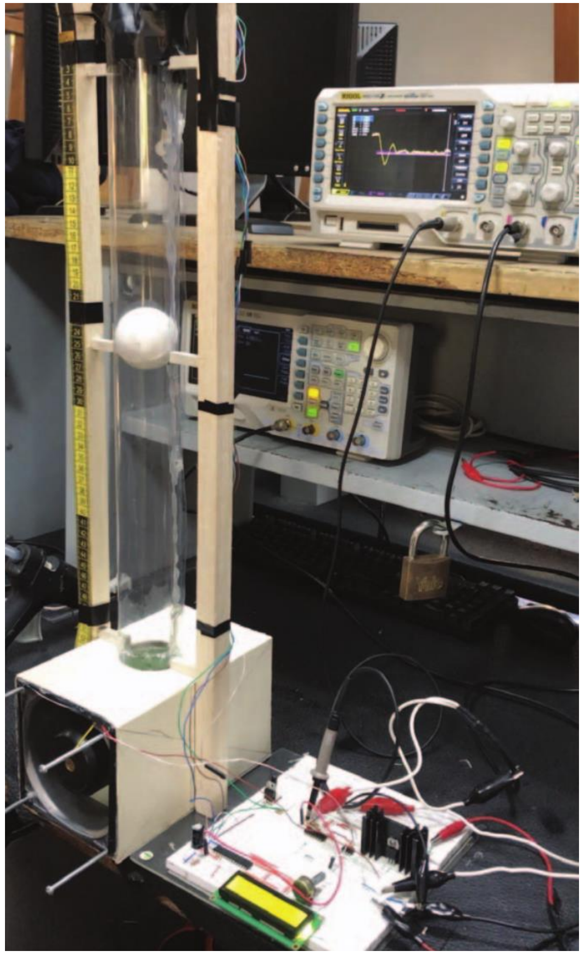
www.automationshield.com

Create novel tools for control engineering and mechatronics education, implementing a lab experiment on a single Arduino expansion Shield, essentially a tiny control / mechatronics laboratory in the palm of your hand that is

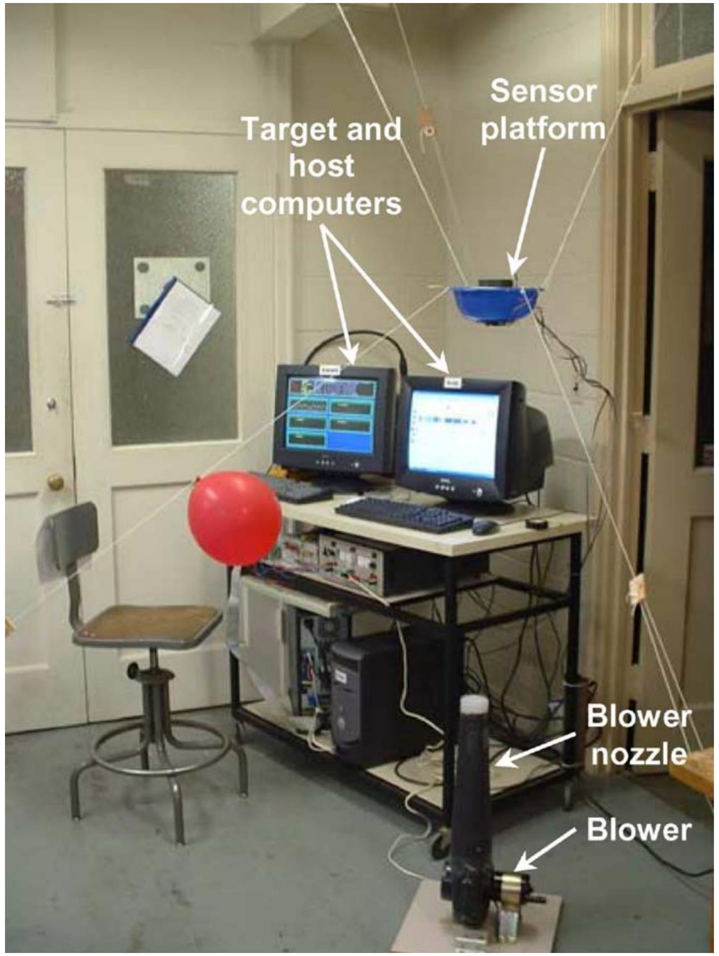
- Cheap
- Open source
- Possible to build at home even by beginners (DIY)
- Standardized
- Free software library compatible with the Arduino IDE (and MATLAB/Simulink)

Motivation: Improvised air flotation devices

Chołodowicz and Orłowski (2017)



Ovalle and Combita (2019)

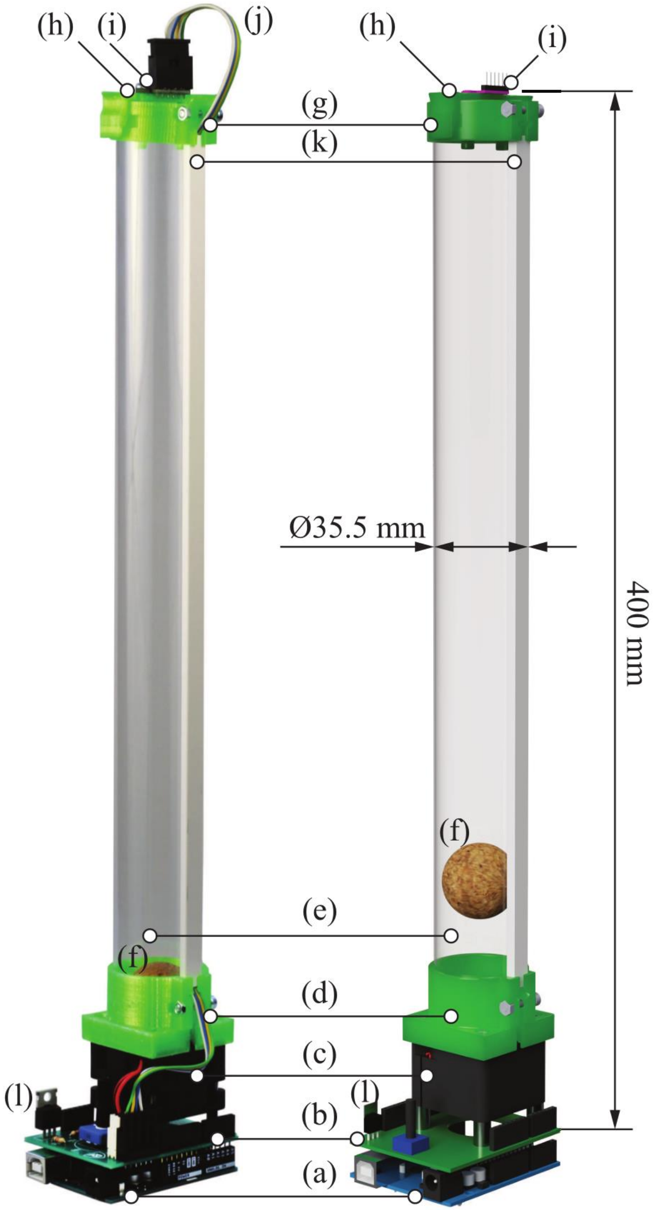


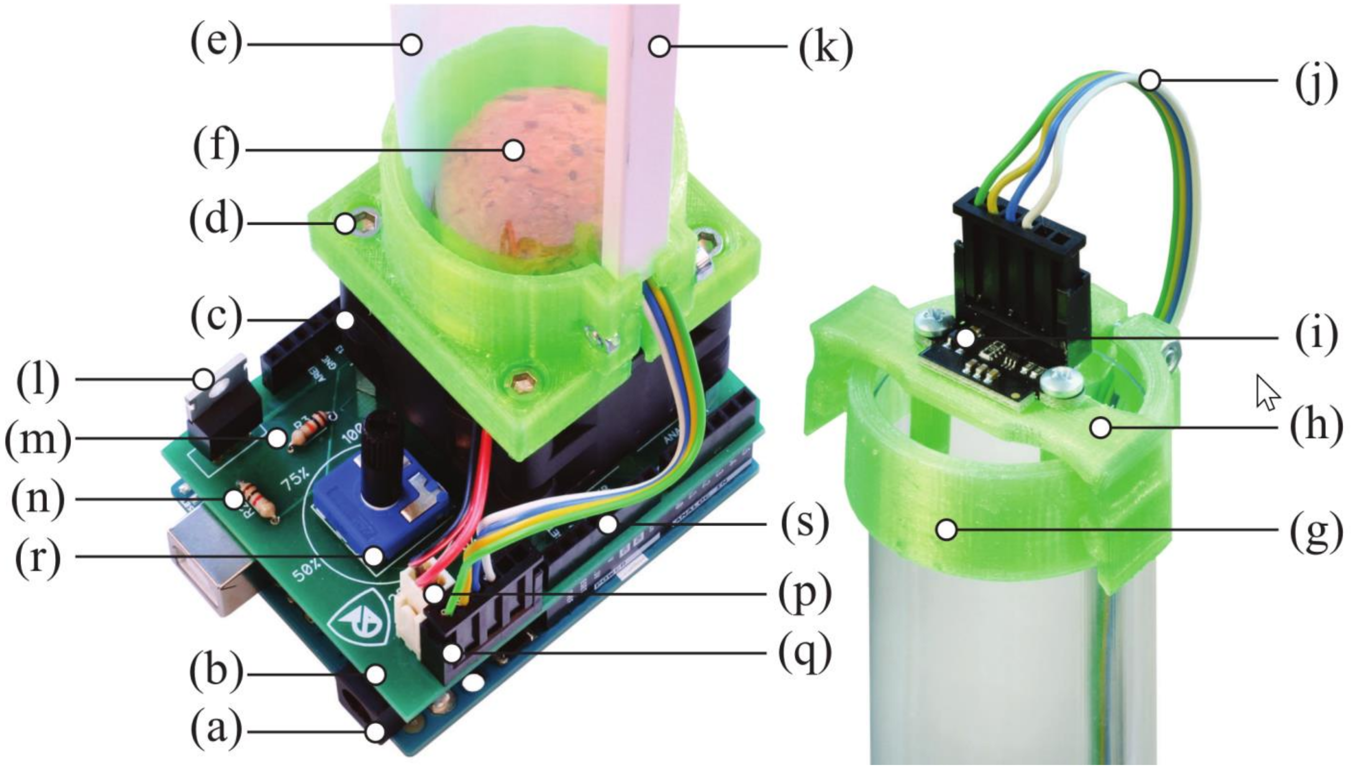
Jernigan et al. (2009)

Motivation: Commercial air flotation devices

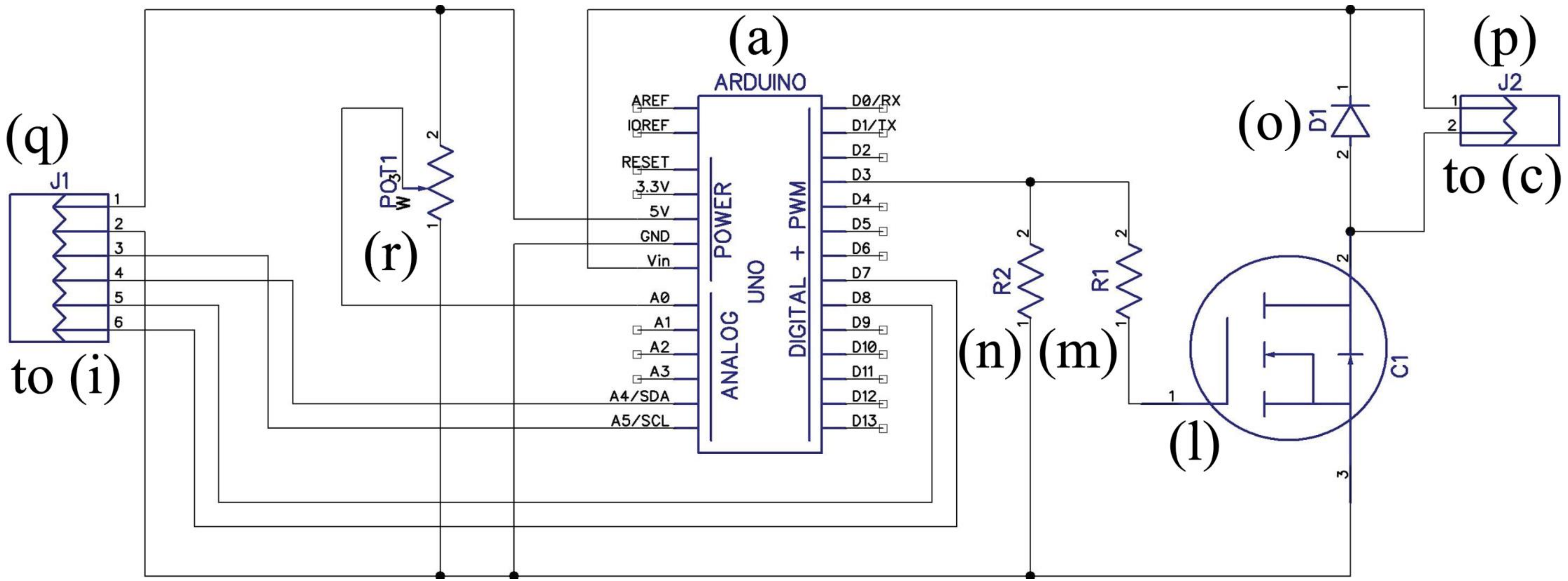
?

FloatShield: Hardware

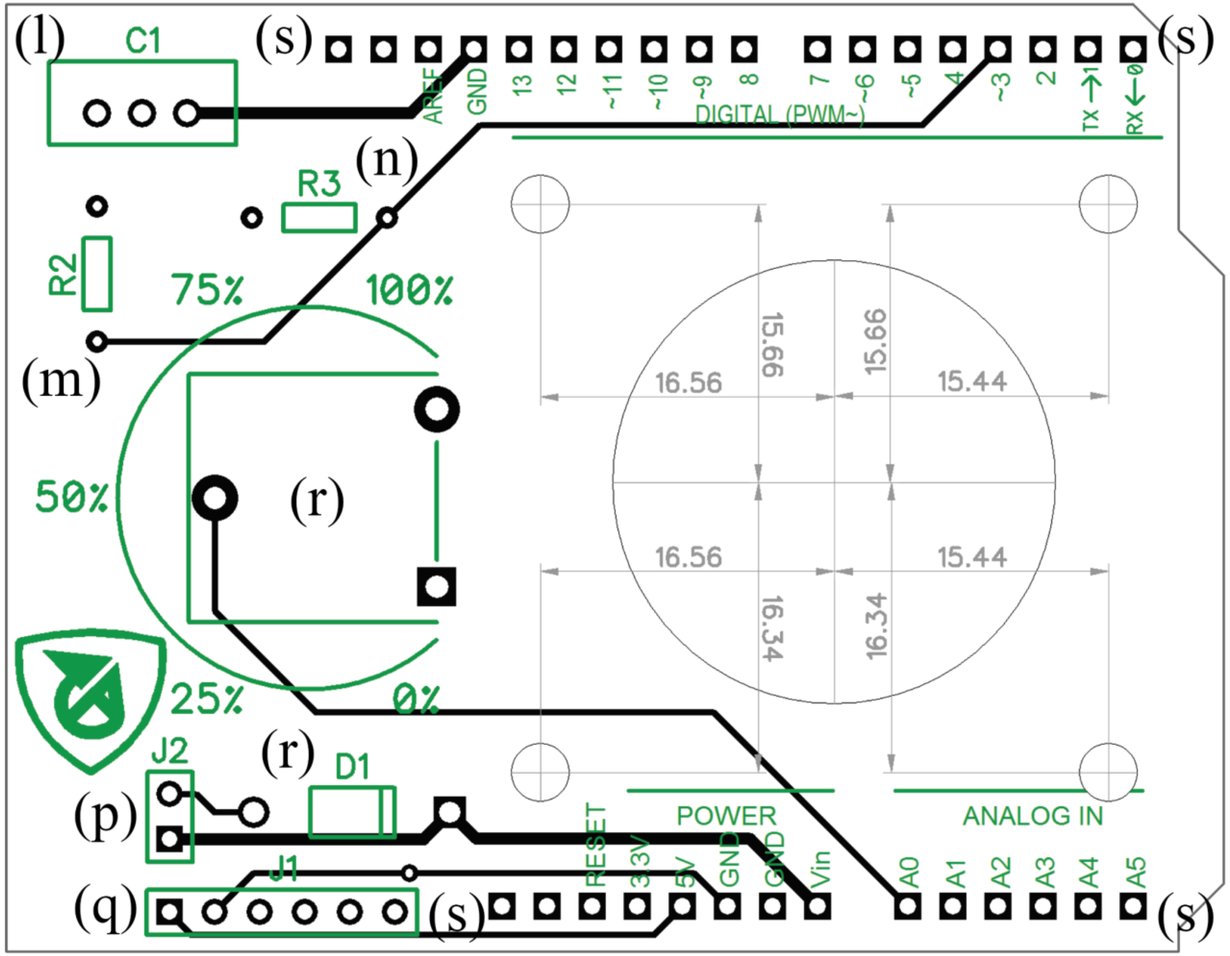


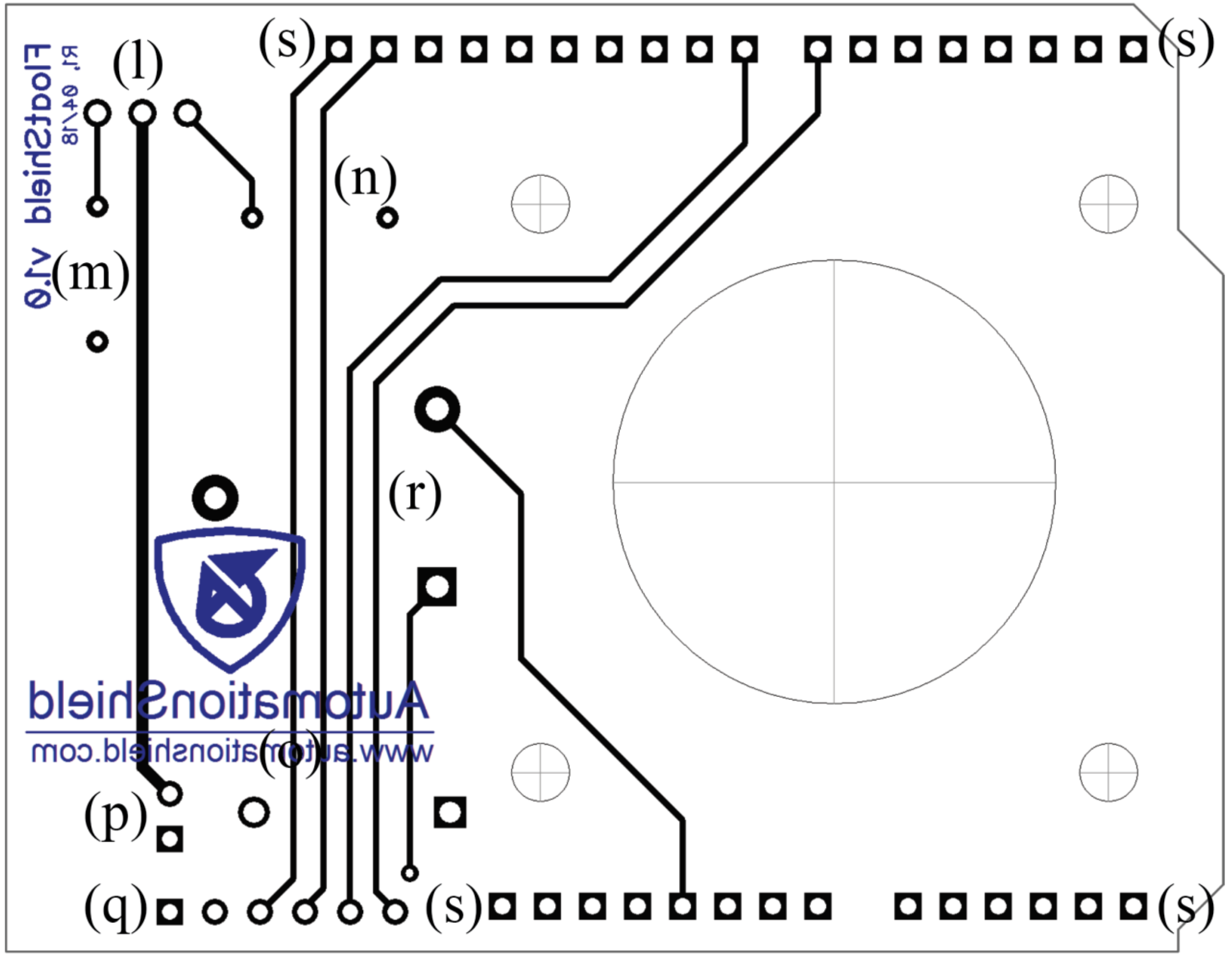


FloatShield: Schematic drawing



FloatShield: Open-source hardware





FloatShield: Component list and price

Symbol	Part	Description	Qty.	UP	Price (€)
(b)	PCB	FR4, 2 layer, 1.6 mm thick	1	0.45	0.45
(c)	Fan	Axial, 12 V, 40×40 mm, 24.0 CFM; e.g. Sunon PMD1204PQB1	1	12.61	12.61
(d)	Tube clamp	3D printed, 16 g filament, print time 2:24 h.	1	0.30	0.30
(e)	Tube	Clear, ϕ35.5 mm, wall approx. 0.6 mm, 0.4 m; e.g. no. 113816	0.4	10.92	4.37
(f)	Ball	Cork, ϕ30 mm; e.g. no. 108269	1	0.63	0.63
(g)	Tube flange	3D printed, 5.8 g filament, print time 57 min.	1	0.11	0.11
(h)	Sensor holder	3D printed, 4 g filament, print time 43 min.	1	0.08	0.08
(i)	Sensor	ST Microelectronics VL5310X TOF sensor on a breakout board	1	5.47	5.47
(j)	Wire	~0.5 m, 4 lead, 0.15 mm^2, multi-conductor ribbon; e.g. VFL 4x0,14	0.5	0.28	0.14
(k)	Cable shaft	U-shape, 8×330 mm, ASA polymer; e.g. 11796	1	1.55	1.55
(l),C1	MOSFET	IRF520, TO-220AB, e.g. IRF520NPBF	1	0.41	0.41
(m), R1	Resistor	1 kΩ , 2.5×6.8 mm, THT	1	0.01	0.01
(n), R2	Resistor	10 kΩ , 2.5×6.8 mm, THT	1	0.01	0.01
(o), D1	Diode	1N4001, e.g 1N4001-DCO	1	0.03	0.03
(p)	Connector (fan)	2×1pin, 0.1" pitch; e.g. TE Connectivity 280358	1	0.04	0.04
(p), J2	Jumper (fan)	2×1pin, 0.1" pitch; e.g. TE Connectivity 280370-2	1	0.16	0.16
(q)	Connector (sensor)	6×1pin, 0.1" pitch; e.g. TE Connectivity 280360	1	0.07	0.07
(q), J1	Jumper (sensor)	6×1 pin, 0.1" pitch; e.g. TE Connectivity 280372-2	1	0.36	0.36
(q),(p)	Connector pins	e.g. TE Connectivity 182206-2	14	0.06	0.90
(r)	Turning knob	5×18.7mm; e.g. ACP 14187-NE	1	0.09	0.09
(r), POT1	Potentiometer	10 kΩ	1	0.28	0.28
(s)	Header	10×1 pin, female, long / stackable, 0.1" pitch	1	0.06	0.06
(s)	Header	8×1 pin, female, long / stackable, 0.1" pitch	2	0.09	0.18
(s)	Header	6×1 pin, female, long / stackable, 0.1" pitch	1	0.09	0.09
-	Bolts	DIN 912 M3×40	4	0.10	0.41
-	Bolts	DIN 912 M3×16	2	0.04	0.08
-	Nuts	DIN 934 M3	6	0.03	0.16
-	Screws	DIN 7981F 2.9×9.5	2	0.03	0.05
-	Washers	DIN125 A 3.2×7×0.5 Polyamide washers	4	0.01	0.03
-	Standoffs	TFM-M3/10	4	0.12	0.46
				Total:	**€ 29.58**[a,b]

Simplified application programming interface (API) in C/C++ included within the **AutomationShield library** for the free Arduino IDE:

- Initialize hardware

```
FloatShield.begin();
```

- Calibrate height reading

```
FloatShield.calibrate();
```

- Read object height to *y*

```
y = FloatShield.sensorRead();
```

- Send a certain power *u* to fan

```
FloatShield.actuatorWrite(u);
```

- Read external reference *r*

```
r = FloatShield.referenceRead();
```

FloatShield: MATLAB API

API available for MATLAB as well, keeps consistent nomenclature and usage with the Arduino API:

- Initialize hardware

```
FloatShield.begin();
```

- Calibrate height reading

```
FloatShield.calibrate();
```

- Read object height to *y*

```
y = FloatShield.sensorRead();
```

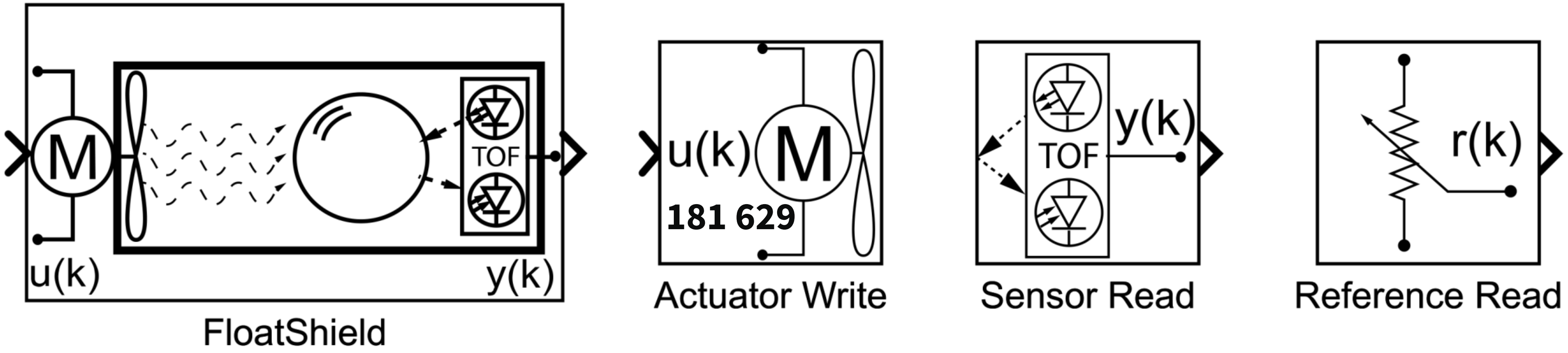
- Send a certain power *u* to fan

```
FloatShield.actuatorWrite(u);
```

- Read external reference *r*

```
r = FloatShield.referenceRead();
```

FloatShield: Simulink API



Typical classroom examples: System Identification

- Modeling
- Data acquisition
- Pre-processing
- Parameter estimation

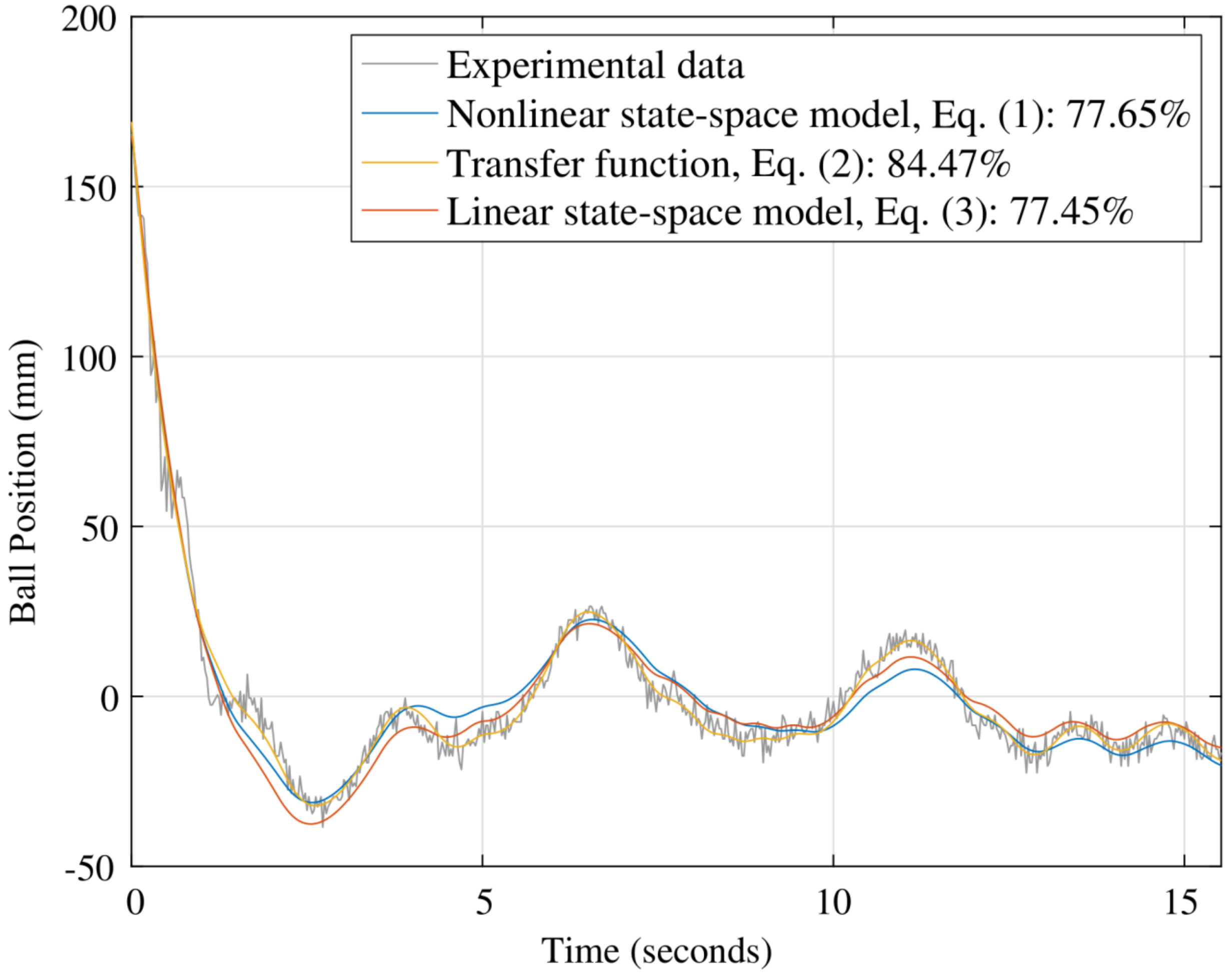
$$\dot{x}_1(t) = x_2(t),$$

$$\dot{x}_2(t) = \frac{1}{2m} c_{\mathrm{d}} \rho A \left(x_3(t) - x_2(t)\right)|x_3(t) - x_2(t)| - g,$$

$$\dot{x}_3(t) = \frac{Ku(t) - x_3(t)}{\tau_1},$$

$$\delta \dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{1}{\tau_2} & \frac{1}{\tau_2} \\ 0 & 0 & -\frac{1}{\tau_1} \end{bmatrix} \delta x(t) + \begin{bmatrix} 0 \\ \frac{K}{\tau_1} \\ 0 \end{bmatrix} \delta u(t)$$

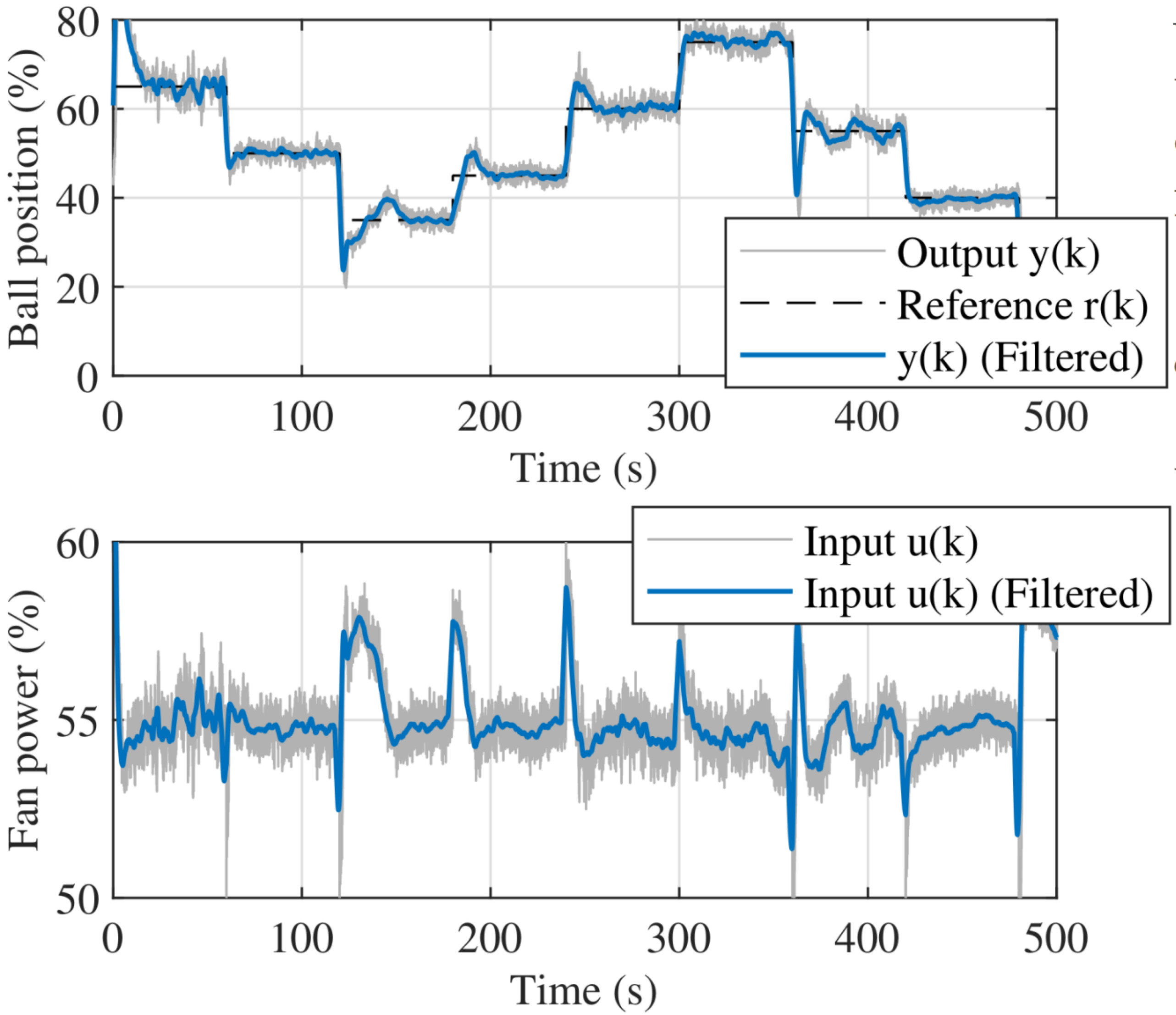
$$\frac{\delta H(s)}{\delta U(s)} = \frac{1}{s(\tau_1 s + 1)(\tau_2 s + 1)},$$



Typical classroom examples: PID control (Arduino IDE)

```
void step() {                                        // Define step function
#if MANUAL                                                                   ve
    r = FloatShield.re                                                       tentiometer
#else                                                                        ctive
    if(i>(sizeof(R)/si                                                       y
        FloatShield.ac
        while(1);
    } else if (k % (T*                                                       on
        r = R[i];
        i++;                                                                 ter
    }
#endif
    y = FloatShield.se
    u = PIDAbs.compute
    FloatShield.actuat

    Serial.print(r);
    Serial.print(" ");
    Serial.print(y);
    Serial.print(" ");
```

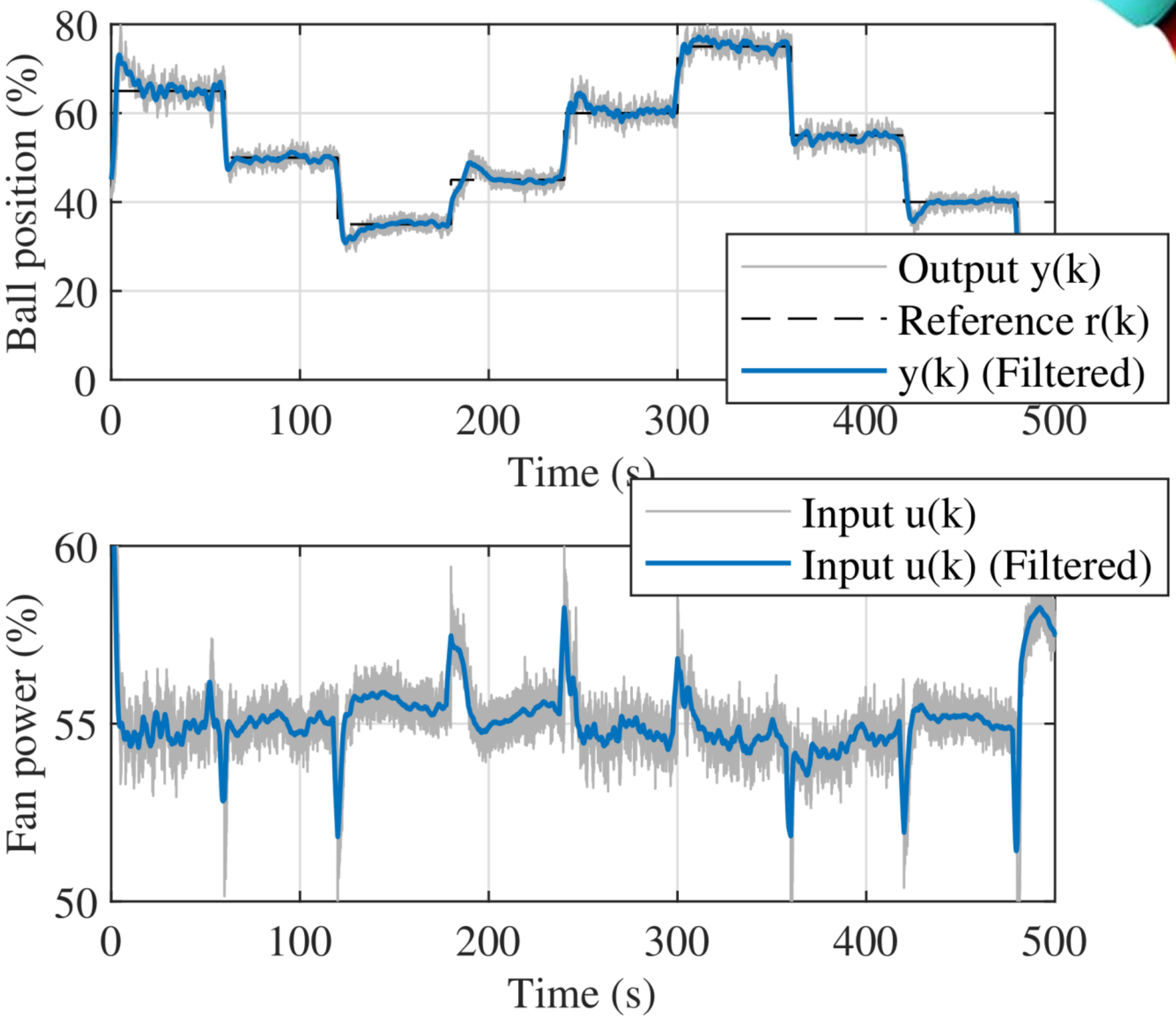


Typical classroom examples: PID control (MATLAB)

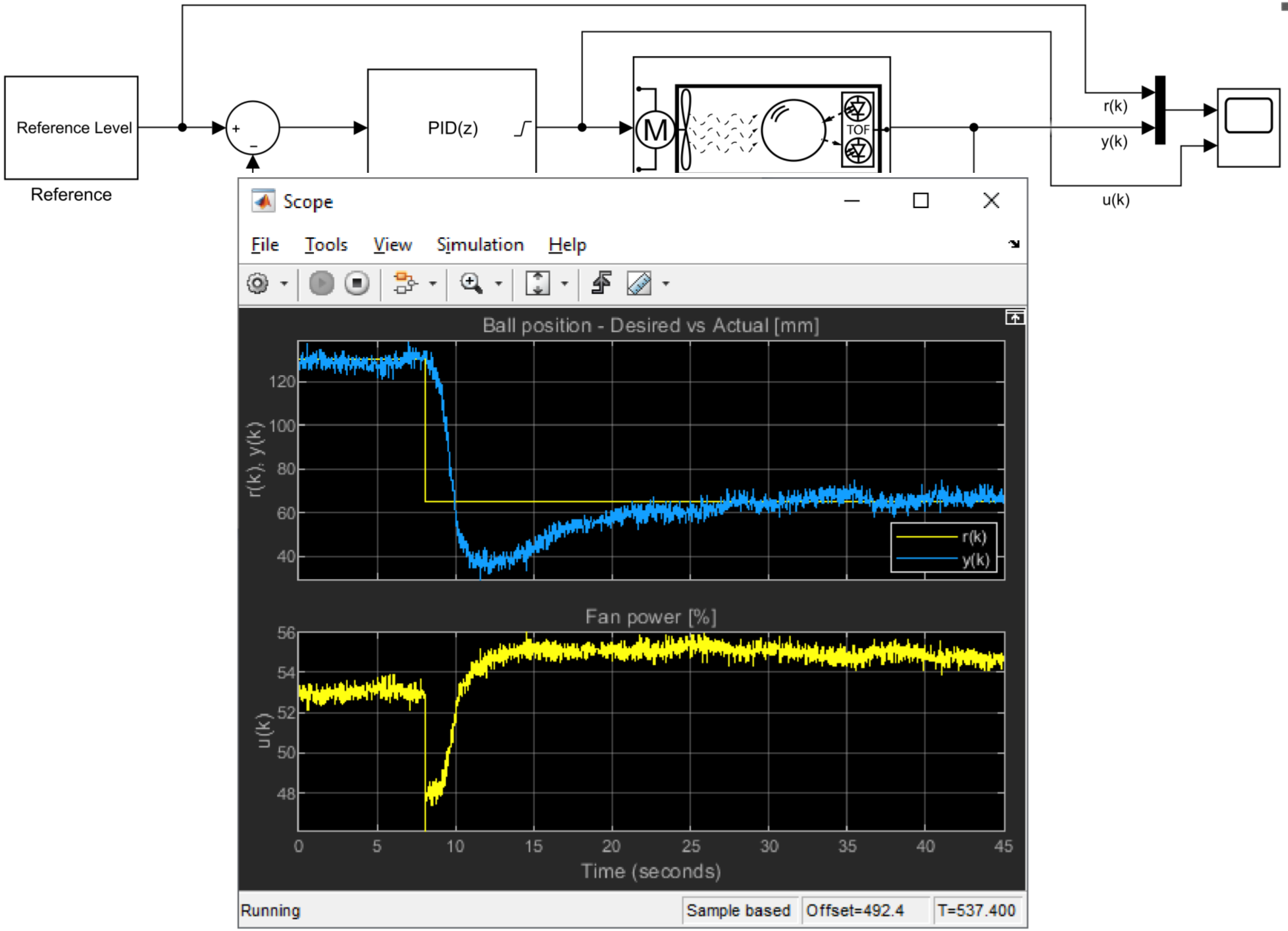
```
while (1)                                        % Infinite loop
    if (nextStep)
        if (mod(k, T*i) == 1)
            i = i + 1;
            if (i > length(R))
                FloatShield.act
                break
            end
            r = R(i);
        end
        y = FloatShield.sensorR
        u = PID.compute(r-y, 0,
        FloatShield.actuatorWri
        response(k, :) = [r, y,
        k = k + 1;
        nextStep = 0;
    end

    if (toc >= Ts * k)
        if (toc >= Ts * (k + 1))
            disp('Sampling viol
            samplingViolation =
            FloatShield.actuato
            break
        end
        nextStep = 1;

end
```



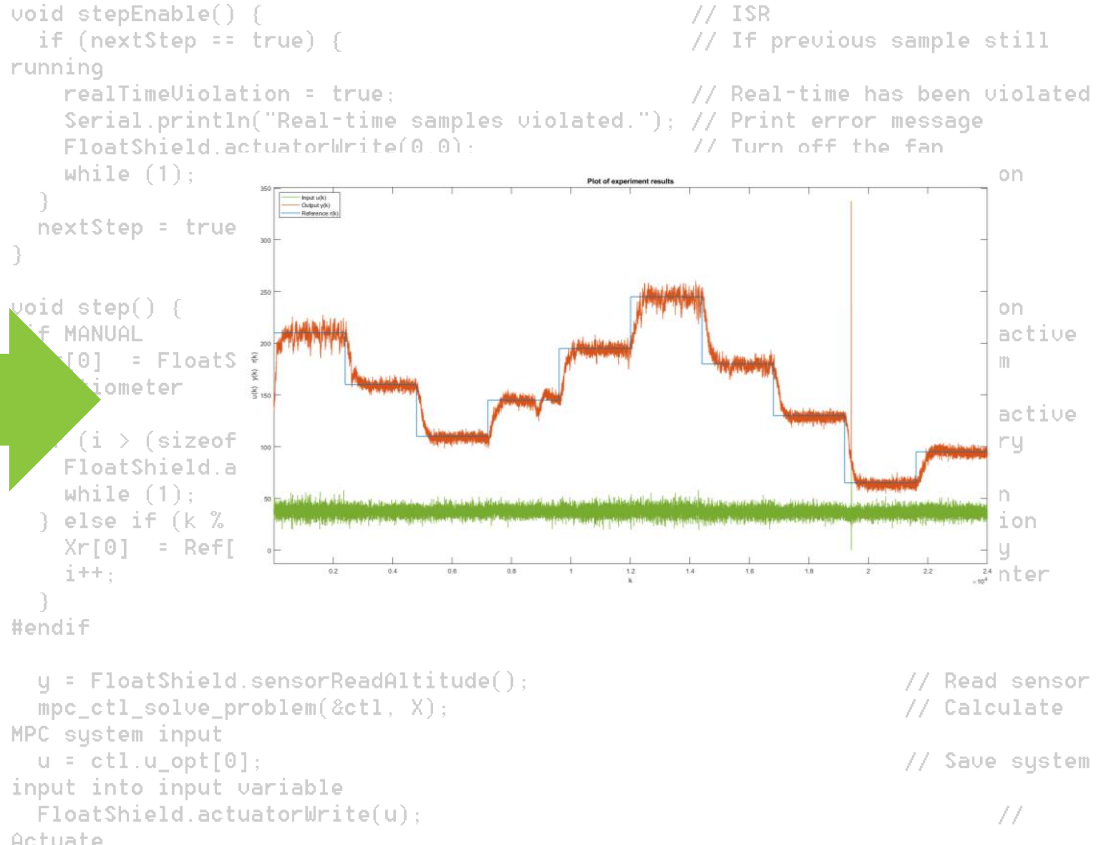
Typical classroom examples: PID control (Simulink)



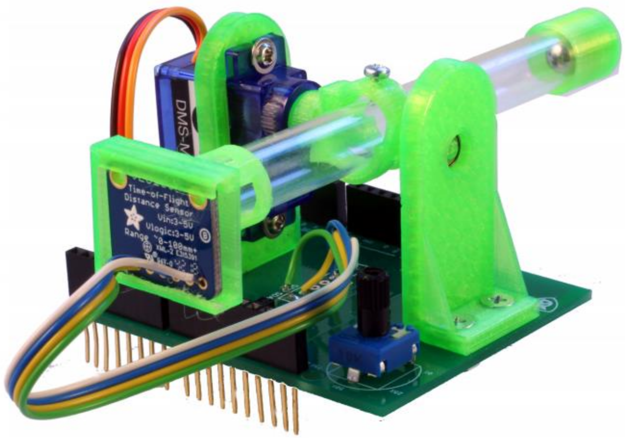
Typical classroom examples: MPC

- Linear MPC running on a 8-bit processor solved by muAO-MPC (Zometa and Findeisen 2016)
- Linear MPC in pseudo real-time solved by “quadprog” (no toolboxes, own implementation in m-files)

$$
\begin{aligned}
\underset{u}{\text{minimize}}\quad & \frac{1}{2}\sum_{j=0}^{N-1}((x_j - x\_ref_j)^T Q (x_j - x\_ref_j) + \\
& (u_j - u\_ref_j)^T R (u_j - u\_ref_j)) + \\
& \frac{1}{2}(x_N - x\_ref_N)^T P (x_N - x\_ref_N) \\
\text{subject to}\quad & x_{j+1} = A_d x_j + B_d u_j, \quad j = 0, \cdots, N-1 \\
& u\_lb \le u_j \le u\_ub, \quad j = 0, \cdots, N-1 \\
& e\_lb \le K_x x_j + K_u u_j \le e\_ub, \quad j = 0, \cdots, N-1 \\
& f\_lb \le F x_N \le f\_ub \\
& x_0 = x
\end{aligned}
$$

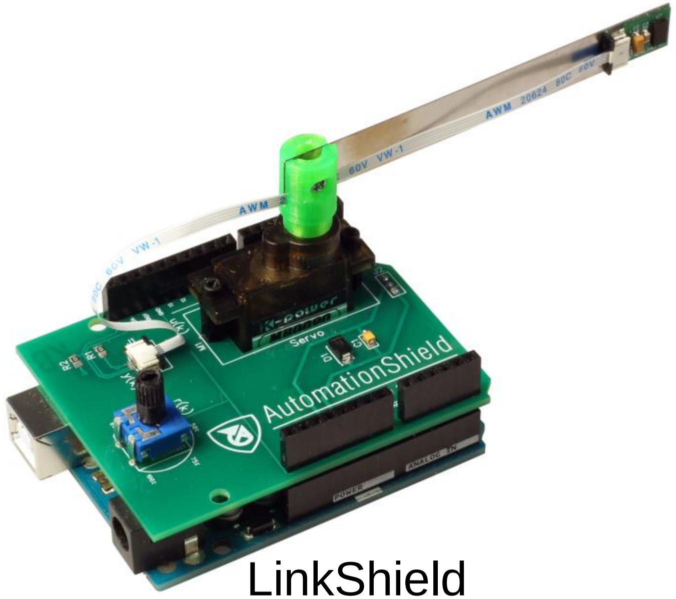


Other shields within our initiative – visit www.automationshield.com

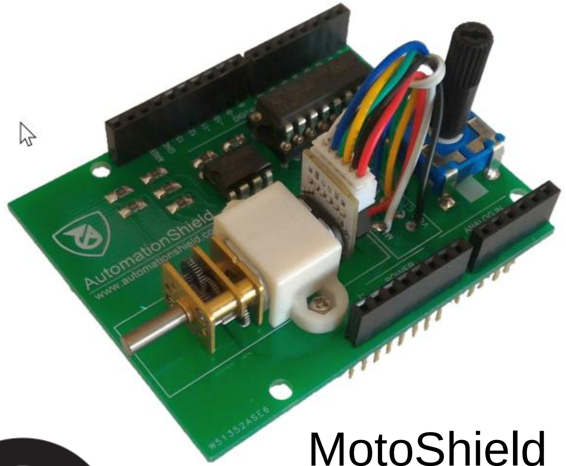


BOBShield (Ball On Beam)

MagnetoShield

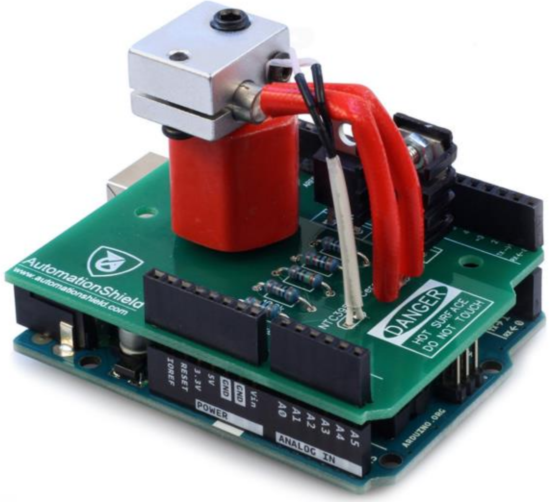


LinkShield



MotoShield

OptoShield



HeatShield

"Take-home" laboratories would be highly desirable for many institutions at this unusual times…

(Several of my students have the "AutomationShield" devices currently at home and thus are a lot less worried about their thesis projects.)

Thank you for your attention!

AutomationShield
Control Systems Engineering Education

Visit www.automationshield.com for more details

and please feel free to contact me any time via:

www: gergelytakacs.com

e-mail: gergelytakacs@gergelytakacs.com

researchgate.net/profile/Gergely_Takacs

linkedin.com/in/gergelytakacs