Control of an air conditioning device by recurrent neural networks

Julius Haas, Prof. Karl Kohlhof

Technology Arts Sciences TH Köln

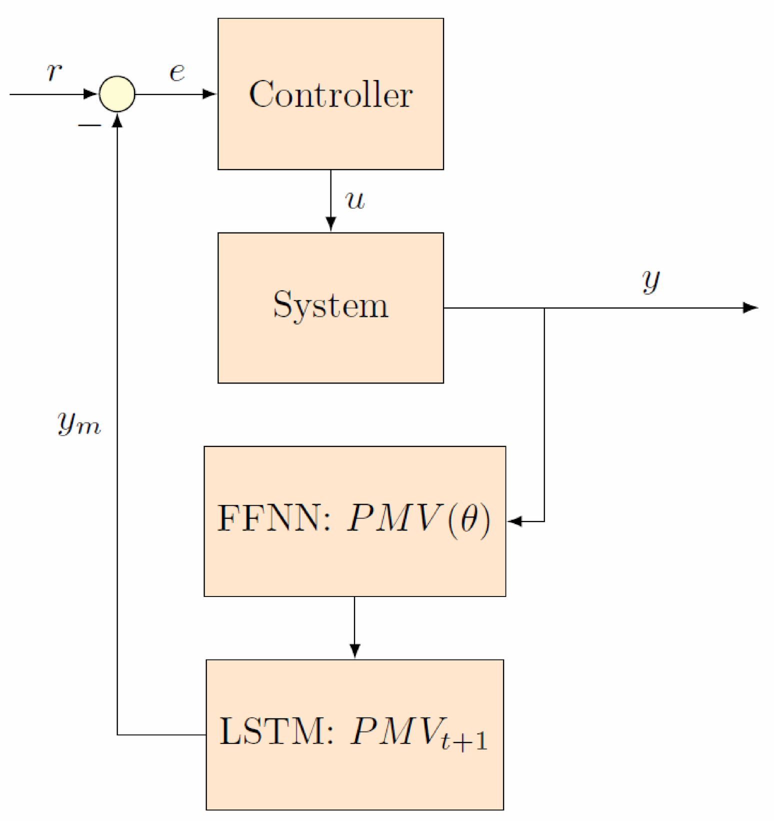
This project is about the creation of training data that is consecutively used to train a reinforcement learning (RL) model including a recurrent neural network (RNN) with the aim of controlling the climate parameters of an interior space while minimizing energy consumption.

Motivation

- Use RL to control thermal comfort of room with PMV input signal and minimum power consumption.
- Room parameters measured by sensor board [1]
- FFNN approximated PMV value for given input signals [2]
- LSTM cell predicts future PMV based on 10 previous PMV values [2]

<u>Scope of this project:</u>

Create training data for the RL algorithm



PMV & PPD

Predicted **M**ean **V**ote (**PMV**) = Empirical value of thermal comfort, based on skin temperature and sweat secretion (thermal equilibrium).

Percentage of **P**eople **D**issatisfied (**PPD**) with room climate. [3]

Defined for: $D_f = \{PMV \in \mathbb{R} | -3 \leq PMV \leq 3\}$

Optimal in: $PMV_{optimal} = [-0.5, 0.5]$

Range of PPD: $PPD = f(PMV) = [0.05, 1]$

Reinforcement learning

Agent (RNN) interacts with environment (Markov decision process (MDP) and gets feedback with state (PMV values) and reward (based on minimal energy consumption).[4]

A *policy* is an agent's strategy: $\pi(s,a) = P(a_t|s_t)$

The *cumulative reward* is the discounted sum of rewards discount:

$$G_t = \sum_{k=0}^{T} \gamma_k \cdot r_{t+k+1} \; (0 \leq \gamma \leq 1)$$

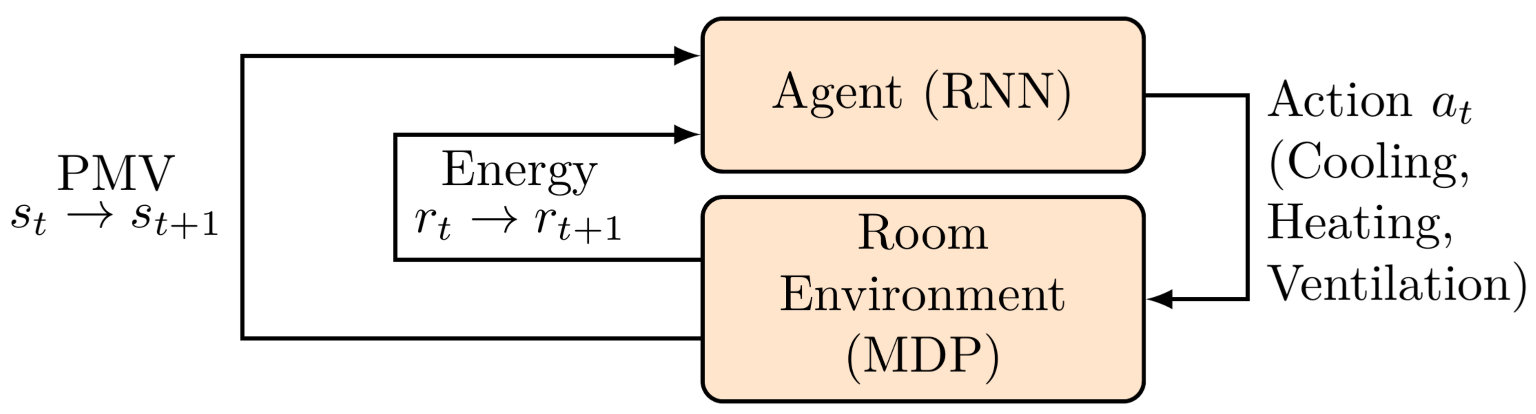
The *state-value function* yields expected return if agent starts in state s and uses the policy to choose its actions for all time steps:

$$V^{\pi}(s) = E_{a\sim\pi}[G_t | S_t = s]$$

The *action-value function* yields expected return if agent starts in state s, takes action a and then follows the policy for all future steps: $Q^{\pi}(s,a) = E_{a\sim\pi}[R_t | S_t = s, A_t = a]$

The *advantage function* determines quality of random action over policy action: $A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s)$

The *gradient descent* represents the fastest way towards a minimum of error function: $\nabla_x f(x) = \sum_{i=0}^{n} \left[\frac{\partial}{\partial x_i}\right] \cdot \hat{e}_i$



Controlling PMV

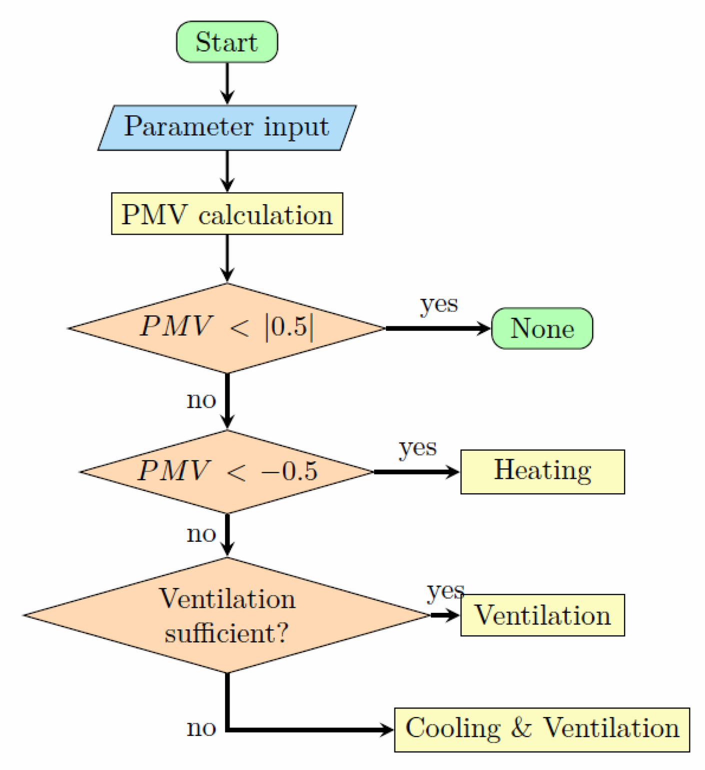
Samsung R32 Maledives is used to control the room climate. Possible actions to control the PMV: heating, cooling, ventilation.

ΔPMV AT 50% HUMIDITY

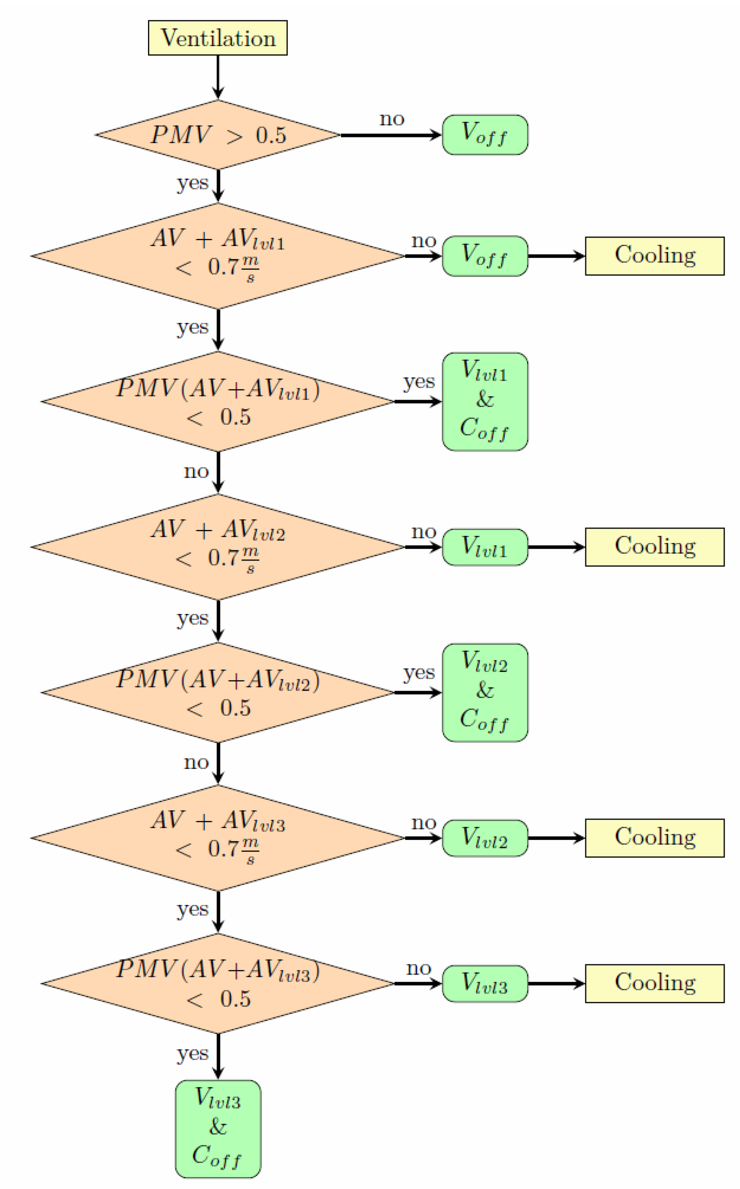
Air velocity [m/s]	16	18	20	22	24	26	28	30
0.45	-0.7	-0.7	-0.6	-0.5	-0.5	-0.4	-0.3	-0.2
0.3	-0.5	-0.5	-0.4	-0.4	-0.4	-0.3	-0.2	-0.2
0.15	-0.2	-0.2	-0.2	-0.2	-0.2	-0.1	-0.1	-0.1
0	0	0	0	0	0	0	0	0

Temperature [°C]

Program sequence



Algorithmic sequence of the training data creation and the ventilation function.



Temperature:

$T = [16°C, 30°C], T_{step} = 0.1K$

Relative humidity:

$H_{rel} = [30\%, 80\%], H_{step} = 1\%$

Air velocity:

$AV = \left[0\frac{m}{s}, 0.7\frac{m}{s}\right], AV_{step} = 0.05\frac{m}{s}$

Training data

id	$T[°C]$	$H_{rel}[\%]$	$AV[\frac{m}{s}]$	PMV_{in}	V_{lvl}	C_{lvl}	H_{lvl}	ΔPMV_{out}	PMV_{out}	PPD_{out}	$P[kW]$
4	16	30	0.15	-2.28	0	0	4	1.9	-0.38	8.01	4
6	16	30	0.25	-2.54	0	0	5	2.47	-0.07	5.1	4.9
12000	17.5	64	0.70	-2.46	0	0	4	2.35	-0.11	5.25	4
12001	17.5	65	0.00	-1.53	0	0	2	1.09	-0.44	9.04	2.2
12004	17.5	65	0.15	-1.7	0	0	3	1.55	-0.15	5.47	3.1
59127	23.7	44	0.55	-0.49	0	0	0	0	-0.49	10.02	0
59128	23.7	44	0.60	-0.52	0	0	1	0.75	0.23	6.1	1.3
59129	23.7	44	0.65	-0.54	0	0	1	0.75	0.21	5.91	1.3
59131	23.7	45	0.00	0.06	0	0	0	0	0.06	5.07	0
76467	25.9	78	0.55	0.52	1	0	0	-0.06	0.46	9.42	0.1
76480	25.9	79	0.45	0.57	1	1	0	-0.53	0.04	5.03	1
80004	26.4	59	0.40	0.59	2	0	0	-0.13	0.46	9.42	0.2
80017	26.4	60	0.30	0.66	2	1	0	-0.5	0.16	5.53	1.1
80031	26.4	61	0.25	0.71	3	0	0	-0.23	0.48	9.81	0.3
80076	26.4	64	0.25	0.73	3	1	0	-0.49	0.24	6.2	1.2
80086	26.4	65	0.00	0.99	3	2	0	-1.23	-0.24	6.2	2.75
105918	29.8	53	0.10	1.78	3	3	0	-1.98	-0.2	5.83	4.3
107859	30	80	0.40	1.99	2	3	0	-2.36	-0.37	7.85	4.2
107860	30	80	0.45	1.98	1	3	0	-2.39	-0.41	8.5	4.1
107861	30	80	0.50	1.97	1	2	0	-1.48	0.49	10.02	2.55
107863	30	80	0.60	1.95	0	2	0	-1.51	0.44	9.04	2.45
107856	30	80	0.25	2.05	3	3	0	-2.27	-0.22	6	4.3
107857	30	80	0.30	2.03	2	3	0	-2.31	-0.28	6.63	4.2
107860	30	80	0.45	1.98	1	3	0	-2.39	-0.41	8.5	4.1
107861	30	80	0.50	1.97	1	2	0	-1.48	0.49	10.02	2.55
107864	30	80	0.65	1.94	0	2	0	-1.52	0.42	8.68	2.45

- Low power consumption for scenarios with only ventilation
- Possibility of power reduction in scenarios where ventilation can substitute or lower cooling levels

Conclusion

Creation of training data successful

- 107864 records of training data created
- Nils Bitzer trained a FFNN with accuracy of 93.3% with the created training data
- Good prerequisite to train RL algorithm offline with the created training data

[1] W. Araar, T. Hofacker, K. Kohlhof, ``Developing an IoT-based control system for existing air conditioner using MEMS,'' University of Applied Sciences Cologne, 2018.
[2] M. Abdi, K. Kohlhof, ``Smart air conditioning according ISO 7730 by neural networks,'' University of Applied Sciences Cologne, 2018.
[3] DIN EN ISO 7730 2005: DIN Deutsches Institut für Normung e.V., 2005.
[4] Goodfellow, Y. Bengio, and A. Courville, ``Deep Learning,'' Massachusetts Institute of Technology, 2016.

Institute of Automation Engineering
Prof. Karl Kohlhof
Institute of Automation Engineering
Karl.Kohlhof@th-koeln.de

Contact: Julius Haas
julius.haas@smail.th-koeln.de
Cologne University of Applied Sciences
(TH Köln)