1. Learning robot and its environment

In this first chapter we briefly discuss the concept behind the learning robot and its environment based on the variable structure stochastic learning automaton (VSLA).
The robot can choose from a finite number of actions (e.g. *drive forwards*, *drive backwards*, *turn right*, *turn left*). Initially at a time $t = n = 1$ one of the possible actions α is chosen by the robot at random with a given probability p. This action is now applied to the random environment in which the robot "lives" and the response β from the environment is observed by the sensor(s) of the robot.

The feedback β from the environment is binary, i.e. it is either favorable or unfavorable for the given task the robot should learn. We define $\beta = 0$ as a reward (favorable) and $\beta = 1$ as a penalty (unfavorable). If the response from the environment is favorable ($\beta = 0$), then the probability p_i of choosing that action α_i for the next period of time $t = n + 1$ is updated according to the updating rule T.
After that, another action is chosen and the response of the environment observed. When a certain stopping criterion is reached, the algorithm stops and the robot has learnt some characteristics of the random environment.

Definition abstract:

- $\underline{\alpha} = \{\alpha_1, \alpha_2, ..., \alpha_r\}$ is the finite set of r actions/outputs of the robot. The output (action) is applied to the environment at time $t = n$, denoted by $\alpha(n)$
- $\underline{\beta} = \{\beta_1, \beta_2\}$ is the binary set of inputs/responses from the environment. The input (response) is applied to the robot at time $t = n$, denoted by $\beta(n)$. In our case, the values for β are chosen to be 0 and 1. $\beta = 0$ represents a reward and $\beta = 1$ a penalty
- $\underline{p} = \{p_1, p_2, ..., p_r\}$ is the finite set of probabilities a certain action $\alpha(n)$ is chosen at a time $t = n$, denoted by $p(n)$
- T is the updating function (rule) according to which the elements of the set $\underline{p}$

are updated at each time $t = n$. Therefore $p(n+1) = \mathrm{T}(\alpha(n), \beta(n), p(n))$, where the i th element of the set $\underline{p}(n)$ is $p_i(n) = \mathrm{Prob}(\alpha(n) = \alpha_i)$ with $i = 1, 2, ..., r$, $\forall\, n: \sum_{i=1}^{r} p_i(n) = p_1(n) + p_2(n) + ... + p_r(n) = 1$ and $\forall\, i: p_i(n=1) = \frac{1}{r}$.

- $\underline{c} = \{c_1, c_2, ...c_r\}$ is the finite set of penalty probabilities that the action α_i will result in a penalty input from the random environment. If the penalty probabilities are constant, the environment is called a *stationary random environment*.

The updating functions (reinforcement schemes) are categorized based on their linearity. The general linear scheme is given by:

If $\alpha(n) = \alpha_i$,

$$\beta = 0: \quad p_j(n+1) = \begin{cases} p_j(n) + a \cdot (1 - p_j(n)) & j = i \\ p_j(n) \cdot (1-a) & \forall\, j \neq i \end{cases}$$

$$\beta = 1: \quad p_j(n+1) = \begin{cases} p_j(n) \cdot (1-b) & j = i \\ \frac{b}{r-1} + p_j(n) \cdot (1-b) & \forall\, j \neq i \end{cases}$$

where a and b are the learning parameter with $0 > a, b < 1$.

If $a = b$, the scheme is called the *linear reward-penalty scheme*. If for $\beta = 1$ p_j remains unchanged ($\forall\, j$ 0), it is called the *linear reward scheme*.