

TD4: Principes des Architectures des Systèmes Autonomes Intelligents

Nicolas Audoux IA-ID, Matthieu Féraud IA-ID

Question 2: Reprenez votre TD précédent et implémentez le déplacement de votre robot en ligne droite et son arrêt devant le premier obstacle rencontré.

Pour s'arrêter devant le premier obstacle, nous avons décidé de regarder à chaque pas de temps la valeurs des capteurs de devants (capteurs 3,4,5 et 6). Si un obstacle est trop proche, le robot s'arrête, sinon il avance tout droit.

```
int main() {
    initialize();

    while (wb_robot_step(time_step) != -1) {
        int i;
        double sensors_value[MAX_SENSOR_NUMBER];

        for (i = 0; i < num_sensors; i++){
            sensors_value[i] = wb_distance_sensor_get_value(sensors[i]);
        }
        if(sensors_value[2]>80 || sensors_value[3]>80 || sensors_value[4]>80
|| sensors_value[5]>80){
            printf("STOP");
            wb_motor_set_velocity(left_motor, 0);
            wb_motor_set_velocity(right_motor, 0);

        }
        else{
            wb_motor_set_velocity(left_motor, max_speed);
            wb_motor_set_velocity(right_motor, max_speed);
        }
    }

    return 0;
}
```

Question 3: Avec le simulateur Webots, en utilisant le robot khepera III, implémentez cet algorithme en utilisant tous les proximètres du robot. Choisissez les bons poids pour que le robot ait un comportement satisfaisant.

On considère équations suivantes avec V_r et V_l les vitesses des moteurs droit et gauche, k une constante, W_{ir} et W_{il} les poids associés aux capteurs i pour le moteur droit et gauche et X_i la valeur du capteur i :

$$V_r = k \cdot \sum_{i=1}^n W_{ir} \cdot X_i$$
$$V_l = k \cdot \sum_{i=1}^n W_{il} \cdot X_i$$

Pour un bon fonctionnement de l'algorithme de Graitemberg, voici les valeurs que nous avons utilisés : $k = 0.00053429$, $X_i = 1.0 - (V_i/2000)$ avec V_i la valeur du capteur i .

Voici la liste des poids pour le moteur droit :

$[-5000, 40000, 50000, 70000, -60000, -40000, -20000, -5000, -10000]$

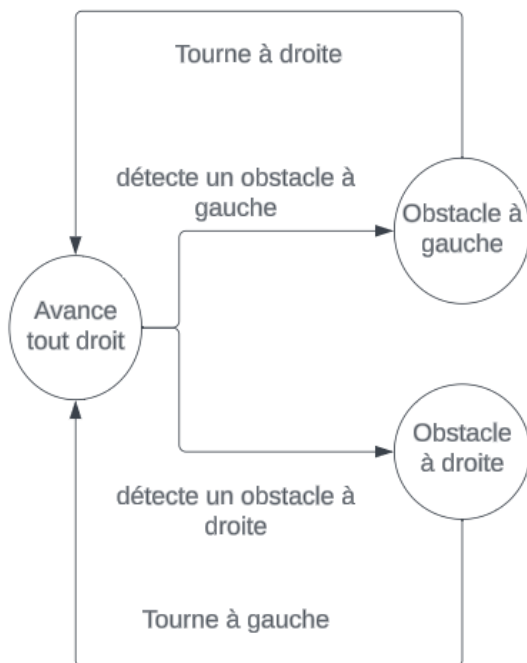
Et celle pour le moteur gauche:

$[-5000, -20000, -30000, -70000, 70000, 50000, 40000, -5000, -10000]$

Question 4: Comment implémenteriez-vous un réseau de neurones pour l'évitement d'obstacles selon braitenberg dans le cas de l'alphabot2 ?

Pour l'alphabot2, il suffirait de trouver les poids correcte pour faire en sorte que lorsqu'un objet est détecté à droite, le robot tourne à gauche (On accélère le moteur droit et on décélère le moteur gauche) et inversement si un objet est détecté à gauche. Le résultat pourrait ne pas être aussi bon que sur le robot kheperas 3 étant donné que l'alphabot ne possède que deux capteurs positionnés à l'avant.

Question 5: Si vous deviez fournir un algorithme d'évitement d'obstacle pour l'Alphabot2, sous forme d'un automate à état fini, quel serait-il ? Implémentez et testez.



Question 6: Quel algorithme mettriez-vous en place pour un suivi de contours d'obstacles par la droite sur le Khepera III ? Implémentez et testez.

Pour suivre les obstacles à notre droite, il faut que notre robot tourne uniquement à gauche. Pour cela, on a modifié les poids de notre matrice pour que le robot tourne toujours à gauche. Notre robot va toujours tout droit est dès qu'un obstacle est détecté, il va tourner à droite en fonction des poids de la matrice. En modifiant encore les poids de la matrice, notre robot pourrait rester plus proche de l'obstacle.

Voici les poids utilisés pour la matrice:

```
const double khepera3_matrix[9][2] = {{-5000, -5000}, {-20000, 40000},  
{-30000, 50000}, {-70000, 70000}, {-60000, 70000}, {-40000, 50000},  
{-20000, 40000}, {-5000, -5000}, {-10000, -10000}};
```

Et voici notre main:

```

int main() {
    initialize();

    while (wb_robot_step(time_step) != -1) {
        int i, j;
        double speed[2];
        double sensors_value[MAX_SENSOR_NUMBER];
        int max_distance = 500;

        if (camera_enabled == 1)
            wb_camera_get_image(camera);

        for (i = 0; i < num_sensors; i++)
            sensors_value[i] = wb_distance_sensor_get_value(sensors[i]);

        for (i = 0; i < 2; i++) {
            speed[i] = 0.0;

            if(sensors_value[2]<max_distance && sensors_value[3]<max_distance
&& sensors_value[4]<max_distance && sensors_value[5]<max_distance){
                speed[0] = max_speed;
                speed[1] = max_speed;
            }

            else{
                for (j = 0; j < num_sensors; j++) {
                    speed[i] += speed_unit * matrix[j][i] * (1.0 - (sensors_value[j]
/ range));
                }
                speed[i] = BOUND(speed[i], -max_speed, max_speed);
            }
        }
        wb_motor_set_velocity(left_motor, speed[0]);
        wb_motor_set_velocity(right_motor, speed[1]);
    }

    return 0;
}

```

Question 7: Si vous voulez implémenter un suivi de ligne grâce aux capteurs du robot Alfabot2, que proposeriez-vous comme algorithme ? Implémentez et testez.

Le robot Alfabot2 possède un récepteur infrarouge. Afin de suivre une ligne noire, une solution serait de faire des droites gauches. Le robot commence à partir à droite et dès que le récepteur a des valeurs supérieures à un seuil, le robot tourne à gauche car cela veut dire qu'il s'éloigne de la ligne noire. On attend ensuite que la valeur du capteur soit à nouveau supérieure au seuil avant de tourner à droite. On répète ensuite ce processus autant qu'il faut. Voici une idée de ce à quoi le programme ressemblerait.

```
int main() {
    initialize();
    int right=1;
    wb_motor_set_velocity(right_motor, max_speed);
    while (wb_robot_step(time_step) != -1) {
        double sensor_value = wb_light_sensor_get_value();
        if (sensor_value>20)
            right = 1-right;

        if (right == 1){
            wb_motor_set_velocity(left_motor, 0);
            wb_motor_set_velocity(right_motor, max_speed);
        }
        else{
            wb_motor_set_velocity(left_motor, max_speed);
            wb_motor_set_velocity(right_motor, 0);
        }
    }

    return 0;
}
```

Lien vers le github:

<https://github.com/Nicolas13210/SIA>