

# AutoRace2025

by vpered tem

# Управление

- Кастомный PID регулятор

```
# ПИД ЛОГИКА
# пропорциональная часть
p_term = kp * error
# интегральная часть
self.integral += error * dt
self.integral = max(min(self.integral, 0.5), -0.5)
i_term = ki * self.integral
# дифференциальная часть
d_term = kd * (error - self.prev_error) / dt
self.prev_error = error
# итоговое управление
steering = p_term + i_term + d_term
```

# Удержание в полосе

```
if found white_line and yellow_line:
```

```
    if white_line_x < left_line_x:
```

```
        swap(white_line, left_line)
```

```
    elif found yellow_line:
```

```
        turn(right)
```

```
elif found white:
```

```
    turn(left)
```

```
else:
```

```
    # use last direction
```

# Детекция знаков + светофор

- Знаки: YOLOv8n + китайский датасет
- Светофор: YOLOE-11n-seg

# Тоннель

- Камера глубины для определения крыши туннеля над нами

```
if self.current_state == self.STATE_LANE:
    if self.ceiling_detected and not found_y and not found_w:
        self.current_state = self.STATE_TUNNEL
        # self.get_logger().info("ВХОД В ТУННЕЛЬ")

    elif self.current_state == self.STATE_TUNNEL:
        # если потолок исчез и мы хоть одну линию видим то
        туннель закончился
        if not self.ceiling_detected and (found_y or found_w):
            self.tunnel_exit_confirmation += 1
        else:
            self.tunnel_exit_confirmation = 0
        if self.tunnel_exit_confirmation >= 3:
            self.current_state = self.STATE_LANE
            # self.get_logger().info("ВЫХОД ИЗ ТУННЕЛЯ")
```

- LIDAR для навигации внутри туннеля

```
if self.current_state == self.STATE_TUNNEL:
    lidar_diff = self.left_dist - self.right_dist
    tunnel_multiplier = 100
    max_tunnel_offset = 90
    offset = np.clip(lidar_diff * tunnel_multiplier,
                    -max_tunnel_offset, max_tunnel_offset)

    # если прижались к стене
    # рулим вправо
    if self.left_dist < 0.30: offset = 100
    # рулим влево
    if self.right_dist < 0.30: offset = -100
    target_x = self.screen_center_x + offset
```