

PROJECT 2025-2026

MoCoM

Auteur principal	Date	Version
PA LAHAROTTE	2025/10/06	V0.01
PA LAHAROTTE	2025/11/24	V0.02

Table des matières

FRENCH VERSION.....	2
Concept général	2
Ressources	2
Livrables attendus (par groupe).....	3
Livrable 1 – Mise en oeuvre d’un contrôle de feu de circulation sous SUMO	3
Livrable 2 – Mise à jour du livrable 1 avec les nouvelles méthodes testées et les nouveaux résultats + Codes	3
Livrable 3 – Communication orale	3
Critères d’évaluation	3
ENGLISH VERSION.....	4
General Concept	4
Resources.....	4
Expected Deliverables (per group).....	5
Deliverable 1 – Implementation of Traffic Light Control in SUMO	5
Deliverable 2 – Update of Deliverable 1 with New Tested Methods and Performances + Codes	5
Deliverable 3 – Oral Presentation.....	5
Evaluation Criteria.....	6

FRENCH VERSION

Concept général

Le projet que vous allez traiter dans le cadre du cours MoCoM vise à vous initier au contrôle de feux de circulation en milieu urbain.

Vous mènerez le projet par groupe de deux, i.e. en binôme.

Le projet prend la forme d'un jeu de rôle, où chaque binôme assure le rôle d'un gestionnaire de réseau routier urbain sur une portion d'un réseau plus large.

!/ ATTENTION : *Les portions de réseau gérées par chacun des binômes sont disjointes, c'est-à-dire que chaque intersection est gérée par un seul et unique gestionnaire.*

En conséquence, il sera demandé à chaque binôme de se positionner sur un ensemble d'intersections à gérer. De fait, chaque binôme sera désigné responsable de la gestion d'un ensemble de quelques intersections (au moins 2 intersections) appartenant à un même réseau.

Chaque binôme pourra alors développer en silo et indépendamment un procédé de gestion des phases des intersections, dont il a la charge. Néanmoins, les codes devront suivre certaines spécifications de façon à être interopérables avec les développements opérés par les autres groupes. Ainsi, il est demandé que l'ensemble des opérations que votre groupe souhaite effectuer à chaque pas de temps soit regroupé dans une fonction Python **groupeXX(.)** appellable à chaque itération de la boucle temporelle de simulation.

Il vous est proposé de travailler depuis un environnement de simulation microscopique du trafic (SUMO) et d'exploiter l'interface TraCI (en Python) pour modifier dynamiquement les phases des feux tricolores. Une initiation aux manipulations de l'environnement de simulation, ainsi qu'à certaines méthodes de contrôle sera proposée sous Python à travers 3 séances de TP. Néanmoins, vous pouvez laisser libre court à votre créativité et votre envie de tester des méthodes de contrôle plus avancées

À l'issue du projet, les procédés de contrôle de chacun des groupes seront rassemblés dans un même fichier exécutable et testés sur le réseau original, de façon à ce que chaque groupe contrôle une portion du réseau routier. Les performances globales seront alors collectées et analysées lors de la séance finale. Dans le même temps, les performances locales telles que collectées individuellement par groupes seront également présentées et défendues par chacun des binômes.

Ressources

- Trois séances de TP
 - TP1 - Initiation à SUMO et au contrôle par feu de circulation
 - Création d'un petit réseau routier 2x2,
 - Modélisation de la demande,
 - Modélisation de feux de circulation,
 - Paramétrage de feux de circulation (contrôle statique),
 - Intégration de TraCI et contrôle dynamique de paramètres,
 - Génération de routines de post-traitement pour analyse des performances.
 - TP2 – Méthodes de contrôle classique – application sur une intersection isolée
 - Mise en œuvre d'une stratégie de contrôle « actuated » et/ou « predicted »
 - TP3 – Méthodes de contrôle avancé – application sur une intersection isolée
 - Mise en œuvre d'une stratégie d'Apprentissage par Renforcement de type Q-Learning
- Documentation SUMO (<https://sumo.dlr.de/docs/>)
- Littérature existante en ligne

Livrables attendus (par groupe)

Livrable 1 – Mise en oeuvre d'un contrôle de feu de circulation sous SUMO

Deadline = 12/01/2026

Le premier livrable vise à évaluer la capacité à prendre en main l'outil de simulation microscopique du trafic SUMO et à comprendre le procédé de modification dynamique des phases de feu. Le livrable se compose des éléments suivants :

- Un repository / dossier avec les codes développés pour
 - intégrer des feux de circulation dans le réseau routier fourni sur les intersections, dont le groupe est responsable,
 - paramétrer un contrôle statique de feux de circulation,
 - créer des fonctionnalités de collecte dynamique d'indicateurs (profil de queue, etc) via TraCi,
 - évaluer la performance multi-critère via des routines de post-traitement
- Un rapport d'étape succinct présentant les travaux effectués, les hypothèses formulées et les performances (multi-critères) obtenues. Ce rapport écrit est l'occasion d'explicitier la démarche adoptée, les hypothèses formulées, les références employées et les performances obtenues suite à l'implémentation

Livrable 2 – Mise à jour du livrable 1 avec les nouvelles méthodes testées et les nouveaux résultats + Codes

Deadline = 30/01/2026

Ce livrable vise à mettre à jour le **rapport** remis en phase 1 en intégrant :

- Un descriptif des méthodes de contrôle testées et implémentées en Python ;
- La logique d'implémentation (justification des choix et hypothèses formulées) ;
- Les performances multi-critères obtenues et les comparer aux approches préalablement implémentées (baseline). L'usage de tableaux comparatifs est vivement recommandé.

En complément, il est attendu la remise d'un **repository/dossier** contenant les codes développés pour effectuer le contrôle des intersections et mentionnant explicitement la fonction groupeXX(.) à exécuter.

Livrable 3 – Communication orale

Deadline = 04/02/2026

Ce livrable vise à présenter les travaux effectués et les performances obtenues par le groupe sur l'ensemble des intersections contrôlées par le groupe, mais également à l'échelle du réseau (en supposant un contrôle fixe sur les autres intersections). Le livrable se compose d'une présentation orale avec support (PPT ou autre) par groupe présentant les intersections contrôlées, les hypothèses formulées, les méthodes de contrôle déployées et les performances obtenues localement (i.e. sur les segments routiers associés aux intersections sous la responsabilité du groupe) et plus globalement en supposant un contrôle fixe sur les autres intersections.

Critères d'évaluation

- Respect des consignes et remise dans les temps impartis des livrables ;
- Qualité des rendus et supports (écrit + oral) et du code ;
- Originalité des facteurs considérés (intégration de perception augmentée (voisinage) / coopération / etc) ;
- Performance locale des algorithmes de contrôle implémentés dans le projet ;
- Participation active au cours et questions posées aux autres groupes.

ENGLISH VERSION

General Concept

The project you will work on as part of the **MoCoM course** is designed to introduce you to **urban traffic light control**.

You will conduct the project in groups of two (i.e., in pairs).

The project takes the form of a **role-playing exercise**, where each pair takes on the role of an **urban road network manager**, responsible for a **specific portion of a larger traffic network**.

/!\ IMPORTANT: The network portions assigned to each pair are **disjoint**, meaning that **each intersection is managed by one and only one group**.

Consequently, each pair will need to **select a set of intersections** to manage. Each group will be **responsible for managing at least two intersections** belonging to the same network. Each pair can then independently and in isolation **develop its own traffic signal control strategy** for the intersections under its responsibility. However, all code must follow certain specifications to ensure **interoperability** between the solutions developed by the different groups.

Specifically, all the operations your group wants to execute at each time step must be **grouped into a single Python function** `groupeXX(.)`, which will be **called at each iteration of the simulation's time loop**.

You will work with the **microscopic traffic simulation environment SUMO** and use the **TraCI interface (in Python)** to dynamically modify traffic light phases. Three practical sessions (TP) will be provided to help you get started with the simulation environment and some basic control methods. However, you are free to **explore more advanced control strategies** if you wish.

At the end of the project, the control strategies developed by each group will be **integrated into a single executable**, and tested simultaneously on the original network, so that **each group controls its own network section**.

Overall performance will then be collected and analyzed during the final session. At the same time, **local performance** (collected individually by each group) will also be presented and defended by each pair.

Resources

- **Three practical sessions (TP):**
 - **TP1 – Introduction to SUMO and Traffic Light Control**
 - Creation of a small 2×2 road network
 - Demand modeling
 - Traffic light modeling
 - Static traffic light control parameterization
 - Integration of TraCI and dynamic control of parameters
 - Generation of post-processing routines for performance analysis

- **TP2 – Classical Control Methods – Application on a Single Intersection**
 - Implementation of an *actuated/predicted control* strategy
- **TP3 – Advanced Control Methods – Application on a Single Intersection**
 - Implementation of a Reinforcement Learning controller, like Q-Learning
- **SUMO Documentation:** <https://sumo.dlr.de/docs/>
- **Existing online literature**

Expected Deliverables (per group)

Deliverable 1 – Implementation of Traffic Light Control in SUMO

Deadline: 12/01/2026

The first deliverable aims to evaluate your ability to use the **SUMO microscopic traffic simulation tool** and to understand the process of dynamically modifying traffic signal phases. The deliverable must include:

- A **repository/folder** containing the code developed to:
 - Integrate traffic lights into the provided road network at the intersections under your group's responsibility
 - Configure static traffic light control
 - Create functions to dynamically collect indicators (e.g., queue profiles) using TraCI
 - Evaluate multi-criteria performance through post-processing routines
- A **short progress report** presenting the work completed, the assumptions made, and the **multi-criteria performance** achieved. This written report should clearly explain the chosen approach, formulated hypotheses, references used, and the results obtained from the implementation.

Deliverable 2 – Update of Deliverable 1 with New Tested Methods and Performances + Codes

Deadline: 30/01/2026

The final deliverable aims to **update the initial report** to include:

- A **description of the control methods** tested and implemented in Python
- The **implementation logic**, including justification of choices and hypotheses
- The **multi-criteria performance** obtained, compared with previously implemented (baseline) approaches. Using **comparative tables** is strongly recommended.

It is also expected to deliver a **repository/folder** containing the code developed to control the intersections, **explicitly indicating the `groupeXX(.)` function** to be executed.

Deliverable 3 – Oral Presentation

Deadline: 04/02/2026

The second deliverable aims to present the work carried out and the performance achieved by the group for all intersections under its control. It is composed of an **oral presentation with visual support** (e.g., PowerPoint or similar) describing:

- The controlled intersections
- The hypotheses made
- The control methods deployed
- The performance achieved **locally** (i.e., on the road segments associated with the group's intersections) and globally (assuming fixed control for the other intersections)

Evaluation Criteria

- Compliance with instructions and deadlines for all deliverables
- Quality of deliverables and presentation materials (written + oral), as well as code quality
- Originality of factors considered (e.g., augmented perception, cooperation, neighborhood integration, etc.)
- **Local performance** of the control algorithms implemented in the project
- Active participation during the course and interaction with other groups