

Guide Téléopération SO-ARM 101

Phase 5 : Téléopération Leader-Follower

Service Écoles-Médias (SEM) - DIP Genève

Prérequis

- Phase 1 complétée (LeRobot installé)
- Phase 2 complétée (Servos configurés avec IDs 1-6)
- Phase 3 complétée (Calibration effectuée)
- Phase 4 complétée (Tests de contrôle validés)
- Scripts SEM installés depuis GitHub
- Environnement lerobot activé

Objectif de la téléopération

Pourquoi la téléopération ? La téléopération permet au robot Follower de reproduire en temps réel les mouvements du robot Leader. C'est l'étape préparatoire essentielle avant l'enregistrement de trajectoires et l'apprentissage par imitation.

La téléopération permet de :

- Contrôler à distance le robot Follower via le Leader
- Tester la synchronisation entre les deux robots
- Valider la configuration avant l'enregistrement
- S'entraîner aux manipulations pour l'apprentissage

Étape 1 : Préparation

Activation de l'environnement

```
# Activer l'environnement conda  
conda activate lerobot  
  
# Se placer dans le dossier des scripts  
cd ~/lerobot/Docs_SEM/scripts  
  
# Vérifier les scripts disponibles  
ls SEM_so101_*teleoperation*.py
```

Vous devriez voir :

- `SEM_so101_5_config_teleoperation.py` - Configuration de la téléopération
- `SEM_so101_6_teleoperation.py` - Mode téléopération temps réel

Configuration matérielle

Élément	Vérification
Leader	Débranché au départ
Follower	Débranché au départ
Alimentations	2x 12V 3A prêtes
Espace de travail	Dégagé pour les deux robots
Calibrations	Fichiers présents (Phase 3 complétée)

 **Important** : Les scripts demandent de brancher les robots un par un dans un ordre précis. **NE PAS brancher les deux robots tout de suite !**

Étape 2 : Configuration de la téléopération

Cette étape utilise le script `SEM_so101_5_config_teleoperation.py` qui va :

- Déetecter les ports USB des robots
- Charger les calibrations de la Phase 3
- Synchroniser les positions initiales
- Tester la communication bidirectionnelle
- Valider la latence et la précision
- Configurer le mode de téléopération (MIROIR ou COPIE)

Lancement du script de configuration

```
python SEM_so101_5_config_teleoperation.py
```

Écran d'accueil



Procédure de configuration en 4 étapes

ÉTAPE 1/4 : Configuration du Leader

```
ÉTAPE 1/4 : CONFIGURATION LEADER

→ Branchez UNIQUEMENT le LEADER
→ Port attendu : /dev/ttyACM0
Appuyez sur ENTRÉE...

✓ Leader détecté sur /dev/ttyACM0
✓ Calibration chargée depuis ~/lerobot/calibration/leader_calibration.json
⌚ Synchronisation des servos...
  Servo 1 (BASE)      : OK (2048)
  Servo 2 (ÉPAULE)    : OK (2048)
  Servo 3 (COUDE)     : OK (2048)
  Servo 4 (POIGNET-F) : OK (2048)
  Servo 5 (POIGNET-R) : OK (2048)
  Servo 6 (POIGNÉE)   : OK (2048)
✓ Leader configuré et centré
```

ÉTAPE 2/4 : Configuration du Follower

```
ÉTAPE 2/4 : CONFIGURATION FOLLOWER

→ Branchez maintenant le FOLLOWER (gardez le Leader branché)
→ Port attendu : /dev/ttyACM1
Appuyez sur ENTRÉE...

✓ Follower détecté sur /dev/ttyACM1
✓ Calibration chargée depuis ~/lerobot/calibration/follower_calibration.json
⌚ Synchronisation avec le Leader...
✓ Follower configuré et synchronisé
```

ÉTAPE 3/4 : Test de synchronisation

Le script effectue automatiquement une série de tests pour valider la communication :

```
ÉTAPE 3/4 : TEST DE SYNCHRONISATION

Test automatique en cours...

Test 1: Servo 1 (BASE)
Leader : 2048 → 1548 → 2548 → 2048
Follower suit : ✓ (latence: 21ms)

Test 2: Servo 2 (ÉPAULE)
Leader : 2048 → 1800 → 2300 → 2048
Follower suit : ✓ (latence: 23ms)

Test 3: Servo 3 (COUDE)
Leader : 2048 → 1700 → 2400 → 2048
Follower suit : ✓ (latence: 22ms)

Test 4: Servo 4 (POIGNET-F)
Leader : 2048 → 1850 → 2250 → 2048
Follower suit : ✓ (latence: 20ms)

Test 5: Servo 5 (POIGNET-R)
Leader : 2048 → 1600 → 2500 → 2048
Follower suit : ✓ (latence: 24ms)

Test 6: Servo 6 (PINCE/POIGNÉE)
Leader : 2048 → 1900 → 2200 → 2048
Follower suit : ✓ (latence: 22ms)

✓ SYNCHRONISATION VALIDÉE
Latence moyenne : 22ms
Précision : ±3 positions
Taux de réussite : 100%
```

ÉTAPE 4/4 : Sélection du mode

```
ÉTAPE 4/4 : SÉLECTION DU MODE

Modes disponibles :
1. MIROIR - Robots face à face (mouvements inversés)
2. COPIE - Robots côte à côté (mouvements identiques)

Votre choix [1/2] : 1

Mode MIROIR sélectionné

Configuration finale :

| Leader      : /dev/ttyACM0
| Follower    : /dev/ttyACM1
| Mode        : MIROIR
| Latence     : 22ms
| Précision   : ±3

✓ Configuration sauvegardée dans :
~/lerobot/calibration/teleoperation_config.json

Vous pouvez maintenant lancer :
python SEM_so101_6_teleoperation.py
```

Comprendre les modes

Mode	Configuration physique	Mouvements	Usage
MIROIR	Robots face à face	Inversés (gauche ↔ droite)	Apprentissage par imitation
COPIE	Robots côte à côté	Identiques	Manipulation synchronisée

🎮 Étape 3 : Mode téléopération

Cette étape utilise le script `SEM_so101_6_teleoperation.py` pour la téléopération en temps réel.

Lancement de la téléopération

```
python SEM_so101_6_teleoperation.py
```

Interface de téléopération

The screenshot shows the terminal window of the teleoperation script. At the top, it displays "SEM - TÉLÉOPÉRATION SO-ARM 101" and "Mode : MIROIR". Below this is a table showing servo positions:

LEADER	FOLLOWER	DIFFÉRENCE
Servo 1: 2048	Servo 1: 2048	0
Servo 2: 2048	Servo 2: 2048	0
Servo 3: 2048	Servo 3: 2048	0
Servo 4: 2048	Servo 4: 2048	0
Servo 5: 2048	Servo 5: 2048	0
Servo 6: 2048	Servo 6: 2048	0

At the bottom of the screen, the following information is displayed:

FPS: 47 | Latence: 21ms | Paquets perdus: 0

COMMANDES :

M : Changer de mode (MIROIR ↔ COPIE)
R : Position repos (centre tous les servos)
P : Pause/Reprendre téléopération
S : Statistiques détaillées
Q : Quitter

[ACTIF] Bougez le Leader pour contrôler le Follower

Utilisation de la téléopération

1. **Manipulation du Leader** : Bougez manuellement les servos du robot Leader. Les servos sont libérés pour permettre le mouvement manuel.
2. **Observation du Follower** : Le robot Follower reproduit instantanément les mouvements. En mode MIROIR, les mouvements sont inversés (gauche devient droite).
3. **Test des limites** : Vérifiez que les limites de calibration définies en Phase 3 sont bien respectées pour les deux robots.
4. **Changement de mode** : Appuyez sur M pour basculer entre MIROIR et COPIE. Le changement est instantané et s'affiche dans l'interface.
5. **Position de repos** : Appuyez sur R à tout moment pour recentrer tous les servos des deux robots à leur position de calibration.

Comprendre l'affichage

Élément	Description	Valeurs normales
LEADER	Position actuelle de chaque servo du Leader	0-4095
FOLLOWER	Position actuelle de chaque servo du Follower	0-4095
DIFFÉRENCE	Écart absolu entre Leader et Follower	< 10
FPS	Fréquence de rafraîchissement	> 30 Hz
Latence	Délai entre mouvement Leader et réponse Follower	< 50ms
Paquets perdus	Erreurs de communication	0

Étape 4 : Tests de validation

Test 1 : Réactivité

1. Bougez rapidement le servo 1 du Leader
2. Le Follower doit suivre avec moins de 50ms de latence
3. Vérifiez l'affichage : Latence < 50ms

Test 2 : Précision

1. Positionnez précisément un servo du Leader
2. Vérifiez dans le tableau que les positions sont identiques (± 5)
3. La colonne DIFFÉRENCE doit rester < 10

Test 3 : Amplitude complète

1. Pour chaque servo, testez les positions MIN → MAX
2. Vérifiez que le Follower atteint les mêmes extrêmes
3. Les limites de calibration doivent être respectées

Test 4 : Mode miroir

1. Appuyez sur M pour activer le mode MIROIR
2. Tournez le servo 1 du Leader vers la gauche
3. Le servo 1 du Follower doit tourner vers la droite

Test 5 : Endurance

1. Laissez la téléopération active pendant 5 minutes
2. Bougez régulièrement les servos
3. Vérifiez qu'il n'y a pas de dérive ou de désynchronisation

Tests réussis si :

- Latence constante < 50ms
- Précision ± 5 positions
- Pas de décrochage après 5 minutes
- Les deux modes (MIROIR/COPIE) fonctionnent

- Pas de paquets perdus (0 erreurs)

 **Dépannage**

Problème	Cause possible	Solution
Script 5 : "Aucun robot détecté"	Robots branchés trop tôt	Débrancher tout, relancer le script, suivre l'ordre
Script 5 : "Port /dev/ttyACM1 non trouvé"	Un seul port USB utilisé	Brancher sur 2 ports USB différents
Script 5 : "Échec du test"	Calibration incorrecte	Refaire la Phase 3 pour les deux robots
Script 6 : "Configuration non trouvée"	Script 5 non exécuté	Lancer d'abord le script 5
Script 6 : "Follower ne suit pas"	Perte de communication	Vérifier câbles, relancer script 5
Script 6 : "Latence > 100ms"	Charge CPU élevée	Fermer autres applications
Script 6 : "Positions décalées"	Mode incorrect	Appuyer sur M pour changer de mode
Script 6 : "Dérive progressive"	Surchauffe des servos	Pause de 5 minutes pour refroidir
Erreur USB : "Permission denied"	Permissions insuffisantes	<code>sudo chmod 666 /dev/ttyACM*</code>
Erreur : "Module not found"	Environnement non activé	<code>conda activate lerobot</code>

Conseils d'utilisation

1. **Ordre de branchement** : Toujours débrancher tout, puis Leader, puis Follower
2. **Ports USB** : Utilisez des ports USB directs sur la carte mère, évitez les hubs
3. **Alimentations** : Vérifiez que les deux alimentations 12V sont stables
4. **Mouvements doux** : Commencez par des mouvements lents pour tester
5. **Surveillance** : Gardez un œil constant sur la latence et les erreurs
6. **Position de sécurité** : Utilisez R pour recentrer en cas de doute
7. **Pause régulière** : Faites des pauses toutes les 10 minutes pour éviter la surchauffe

Comprendre les fichiers de configuration

Fichiers créés par les scripts :

Fichier	Créé par	Contenu
~/lerobot/calibration/leader_calibration.json	Phase 3	Limites MIN/MAX/CENTRE de chaque servo Leader
~/lerobot/calibration/follower_calibration.json	Phase 3	Limites MIN/MAX/CENTRE de chaque servo Follower
~/lerobot/calibration/teleoperation_config.json	Script 5	Ports USB, mode, paramètres de synchronisation

Commandes de référence rapide

```
# Installation (si pas déjà fait)
cd ~/lerobot
git clone https://github.com/yanko-sem/SO-ARM-101.git Docs_SEM

# Configuration initiale (Script 5)
conda activate lerobot
cd ~/lerobot/Docs_SEM/scripts
python SEM_so101_5_config_teleoperation.py

# Lancement téléopération (Script 6)
python SEM_so101_6_teleoperation.py

# Commandes clavier dans le script 6
M : Basculer mode MIROIR/COPIE
R : Position repos (centre tous les servos)
P : Pause/Reprendre
S : Statistiques détaillées
Q : Quitter proprement

# En cas de problème USB
sudo chmod 666 /dev/ttyACM*

# Vérifier les ports disponibles
ls /dev/ttyACM*
```

Notes finales

Phase 5 terminée quand :

- Les deux robots sont détectés automatiquement
- La configuration (script 5) se termine sans erreur
- Le Follower suit le Leader en temps réel
- La latence reste inférieure à 50ms
- Les deux modes (MIROIR/COPIE) fonctionnent
- Vous maîtrisez les commandes de contrôle

 **Objectif atteint :** Votre système de téléopération est maintenant opérationnel ! Les robots peuvent travailler ensemble pour l'enregistrement de trajectoires et l'apprentissage par imitation qui seront couverts dans la Phase 6.

Service Écoles-Médias - DIP Genève

Guide Phase 5 - Version 1.0

17.12.2025