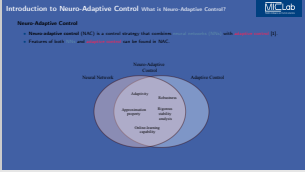
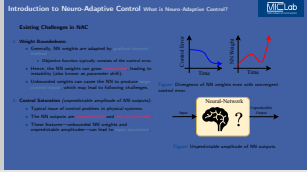


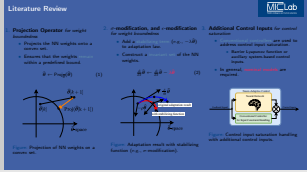
1. adaptive controls adapts their parameters in real-time to systems.
   - Hence, they can handle uncertainties and disturbances.
2. Like adaptive control, NAC adapts its NN weights, as well.
3. Hence, it has both features of NN and adaptive control.

Figure: General framework of neuro-adaptive control (NAC).

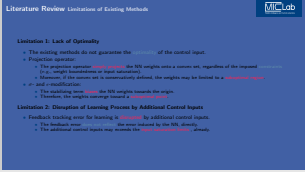
1. The features are...
   - It can adapts its weights to change environments.
   - And the stability is ensured in the sense of Lyapunov.
   - So, it has online learning capability.
   - In addition, with various methods from adaptive control, NAC can be more robust than naive NN-based control methods.



1. However, there are some challenges in NAC.
2. The first challenge is weight boundedness.
   - If the NNs are used for control naively, the boundedness of NN weights is not ensured.
   - For instance, despite the convergence of the control error, the NN weights can diverge like parameter drift.
   - This divergence leads to the following issue.
3. Second one is the NN's output can be excessively large control inputs.
   - This is because, the output of NN is not interpretable and unpredictable.
   - So, the input saturation problem can occur by them.

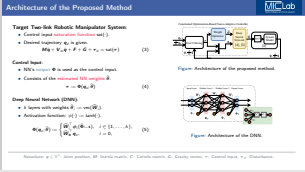


1. To address these issues, several methods have been introduced.
2. For weight boundedness, the projection operator and modification methods are used.
   - they ensures the weight boundedness by projects onto a convex set or adding a stabilizing term.
   - Hence, the the boundedness of NN weights can be easily ensured.
3. For control input saturation, additional control inputs are used.
   - Conventionally, auxiliary system-based control inputs or, more recently, barrier Lyapunov function-based control inputs are used.
   - For this, the nominal model knowledge is required.
   - These makes the control design more complicated, but, have shown the effectiveness of the control input saturation handling.

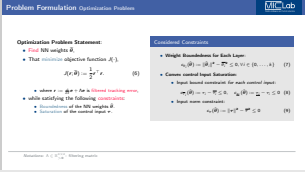


1. the limitations of the existing methods can be summarized as follows.
2. Most of methods does not consider the optimality when the constraints are active.
   - Because, they only consider the boundedness of the weights or the control input saturation using additional operators or terms.
3. Using additional terms can lead to disruption of learning process.
4. It is, because, as the NN learns with respect to the feedback error, once the additional control inputs are applied, the feedback error does not reflect the error induced by the NN, directly.
5. Moreover, the additional control inputs may exceed the input limits, already.
6. Therefore, in the worst case, the NN may be forced to compensate for the additional terms.

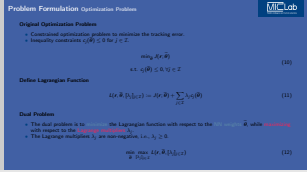
1. So, here are the contributions of this work.
2. We, first, formulated a unified optimization framework for trajectory tracking and constraint handling.
3. Hence, we do not need the additional operators or conventional controllers, which may need the nominal model knowledge.
4. Second, we rigorously proved the stability of the proposed method using Lyapunov stability theory.
5. It means, that the proposed method can learn online without any prior system knowledge with stability guarantees.
6. Finally, as we explicitly consider the weight and control input constraints in the optimization problem, we can handle any combination of convex input constraints and weight constraint.



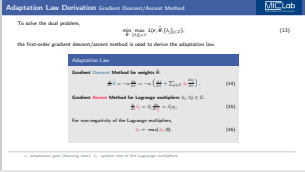
1. The proposed method consists of a DNN and weight optimizer, inside.
2. Let's start with the system.
   - The system is a two-link robotic manipulator system, which is controlled by the control input $\boldsymbol{\tau}$.
   - The control input is saturated by the saturation function sat$(\cdot)$.
   - The desired trajectory $\boldsymbol{q}_d$ is given.
3. The control input is purely decided by the DNN, so the output of the DNN is the control input.
4. And, the DNN is defined like (5), with k layers and weights, and $\tanh(\cdot)$ as the activation function.



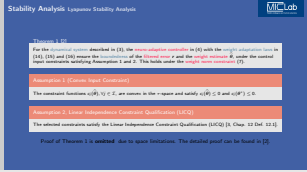
1. The control problem can be represented as a constrained optimization problem.
   - That is,
   - We need to find NN weights ...
2. To explicitly consider the constraints, we define the constraint functions as follows, for weight boundedness and convex control input saturation of each control input and norm of the control input.



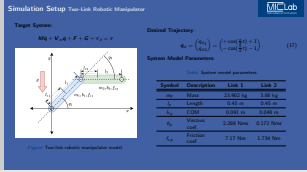
1. Then using the constraint functions, we can formulate the constrained optimization problem as in (10).
2. And, we can define the Lagrangian function as in (11).
3. So, the dual problem can be formulated as in (12).
4. It means, that we can find the optimal solution by minimizing the Lagrangian function with respect to the NN weights, while maximizing it with respect to the Lagrange multipliers.



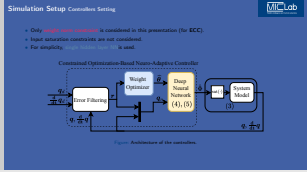
1. To solve this problem, we employed the first-order gradient descent/ascent method.
2. For minimizing the Lagrangian function with respect to the NN weights, the gradient descent method is used.
   - We have a adaptation gain α to control the learning rate.
3. For maximizing the Lagrangian function with respect to the Lagrange multipliers, the gradient ascent method is used.
   - We have an update rate β_j for each Lagrange multiplier.
4. Finally, the max operator is used to ensure the non-negativity of the Lagrange multipliers.
5. Therefore, according to the adaptation law, when the constraints are active, the Lagrange multipliers are updated to be positive.
6. And, the NN weights are updated to minimize the constraint violation while minimizing the tracking error.
7. After the constraints are satisfied, *negative, the Lagrange multipliers are updated to zero, and the NN weights are updated to minimize the tracking error only.*



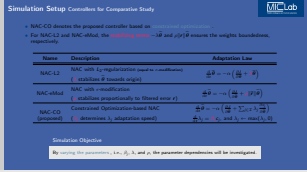
1. The stability of the proposed methods is proven using Lyapunov stability theory.
2. The requirements for the stability are as follows.
3. First, the weight norm constraints should be imposed.
4. And, two assumptions for convex input constraints and LICQ conditions should be satisfied.
5. The proof is omitted, but the detailed proof can be found in the reference [2].



1. For validation, we simulated a two-link robotic manipulator system.
2. A simple cosine trajectory is given as the desired trajectory.



1. For this conference paper, we only considered the weight norm constraint.
2. So, please ignore the input saturation.



1. For comparative study, we used two existing methods: L2-regularization (NAC-L2) and ϵ-modification (NAC-eMod).
2. Please, note that the L2-regularization is equal to the σ-modification.
3. In summary, the NAC-L2 and NAC-eMod use the stabilizing terms $-\lambda\widehat{\boldsymbol{\theta}}$ and $\rho\|\boldsymbol{r}\|\widehat{\boldsymbol{\theta}}$, respectively, to ensure the boundedness of the weights.
4. While the NAC-CO is the proposed method, which uses the Lagrange multipliers λ_j to ensure the boundedness of the weights.
5. The simulation is mainly conducted to investigate dependencies of the crucial parameters, i.e., β_j, λ, and ρ.

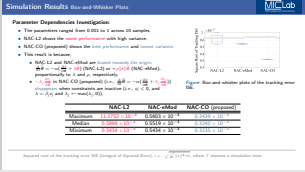
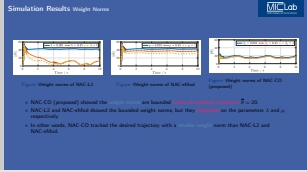


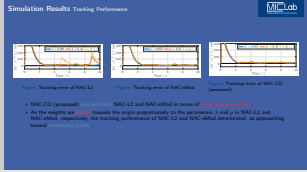
Figure: Box-and-whisker plots of the tracking error ISE.

Squared root of the tracking error ISE (Integral of Squared Error), $i.e., \sqrt{\int_0^{t_f} (e)^2 dt}$, where t_f denotes a simulation time.

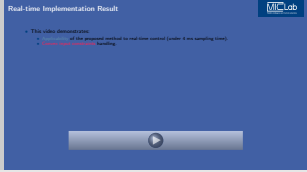
1. The parameters were varied from 0.001 to 1 across 10 samples.
2. The box-and-whisker plots show the distribution of the tracking error ISE (Integral of Squared Error).
3. The NAC-L2 shows the worst performance with high variance, while the NAC-CO (proposed) shows the best performance with the lowest variance.
4. The reason is that the NAC-L2 and NAC-eMod are biased towards the origin, especially, proportionally to the parameters λ and ρ, respectively.
5. On the other hand, the term that biases the weights towards the origin in NAC-CO, is removed when the constraints are inactive, i.e., $c_j < 0$.
6. Therefore, the NAC-CO can track the desired trajectory without being biased, when the constraints are inactive.



1. In this slide, the weight norms are shown.
2. We can see that the NAC-CO's weights are bounded under the pre-defined constraint $\overline{\theta} = 20$.
3. The other two methods shows larger weight norms, and biased weights towards the origin, which may lead to suboptimal performance.
4. Moreover, the dependencies also can be seen, as the distribution of the weight norms of NAC-L2 and NAC-eMod are wider than that of NAC-CO.



1. The tracking performance of the three methods are shown.
2. The NAC-CO (proposed) outperformed the other two methods in terms of tracking performance.
3. As the weights are biased towards the origin proportionally to the parameters λ and ρ in NAC-L2 and NAC-eMod, respectively, the tracking performance of NAC-L2 and NAC-eMod deteriorated, as approaching toward suboptimal points.



1. And, until the last slice, we have shown the conference paper's results.
2. In this slide, we show the recent result with real-time implementation.
3. The video demonstrates the applicability of the proposed method to real-time control, under 4 ms sampling time.
4. And, the convex input constraints are handled, here.

Temporary page!

LaTeX was unable to guess the total number of pages correctly. As there was some unprocessed data that sh
have been added to the final page this extra page has been added to receive it.

If you rerun the document (without altering it) this surplus page will go away, because LaTeX now knows ho
pages to expect for this document.