Legend		
Math	Description	In code
[ullet imesullet]	Matrix	[•,•]
[1 imes ullet]	vector or 1-dim. array	[•,]
r	Running cost	rcost
N_a	Actor control horizon	Nactor
N_c	Critic stack size	Ncritic
N_m	Model estimator stack size	modEstBufferSize
J	Controller cost function	actorCost
J_c	Critic cost function	criticCost
Q,\hat{Q}	Q-function and its approximate	
e	Temporal difference	е
W,W^-	Current and previous critic weights	W, Wprev
R_1,R_2	Parameters of r	R1, R2
γ	Discounting factor	gamma
L	Number of critic weights	dimCrit
Δt	Controller sampling time	dt
δ	Prediction step size	predStepSize
t_0,t_1	Start time and total time of one episode	t0, t1
	Number of episodes	Nruns
x, n	State and its dimension	x, dimState
u, l	Control input and its dimension	u, dimInput
y, p	Output and its dimension	y, dimOutput
\hat{x},\hat{n}	Estimated model state and its dimension	-, modelOrder

Controller mode Ctristatmode		
0 - Manual control		
10 - Nominal control		
1, 2 - Model-predictive control (MPC)	$J\left(y_1,\left\{u ight\}_1^{N_a} ight)=\sum_{k=1}^{N_a}r(y_k,u_k)$	
3, 4 - RL/ADP via N_a-1 roll-outs of \emph{r}	$J\left(y_{1},\left\{ u ight\} _{1}^{N_{a}} ight) =\sum_{k=1}^{N_{a}-1}r(y_{k},u_{k})+\hat{Q}(y_{N_{a}},u_{N_{a}})$	
5, 6 - RL/ADP via normalized stacked Q-learning	$J\left(y_{1},\left\{ u ight\} _{1}^{N_{a}} ight)=rac{1}{N_{a}}\sum_{k=1}^{N_{a}-1}r(y_{k},u_{k})+\hat{Q}(y_{N_{a}},u_{N_{a}})$	

Modes 1, 3, 5 use true model (f,h) for prediction Modes 2, 4, 6 use a state-space model estimated online





