

# Inverted Pendulum Transporter

Generated by Doxygen 1.8.9.1

Sun Dec 11 2016 18:07:19



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List	1
<b>2</b>	<b>Hierarchical Index</b>	<b>3</b>
2.1	Class Hierarchy	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List	5
<b>4</b>	<b>Namespace Documentation</b>	<b>7</b>
4.1	Package InvertedPendulumTransporter	7
4.2	Package InvertedPendulumTransporter.Controls	7
4.3	Package InvertedPendulumTransporter.Models	7
4.4	Package InvertedPendulumTransporter.Properties	8
4.5	Package InvertedPendulumTransporter.Windows	8
4.5.1	Enumeration Type Documentation	8
4.5.1.1	WindowType	8
4.6	Package InvertedPendulumTransporterPhysics	8
4.7	Package InvertedPendulumTransporterPhysics.Common	9
4.8	Package InvertedPendulumTransporterPhysics.Controllers	9
4.8.1	Enumeration Type Documentation	10
4.8.1.1	AccuracyType	10
4.8.1.2	ControlType	10
4.8.1.3	WindType	10
4.9	Package InvertedPendulumTransporterPhysics.Solvers	10
<b>5</b>	<b>Class Documentation</b>	<b>11</b>
5.1	InvertedPendulumTransporter.Windows.AboutWindow Class Reference	11
5.1.1	Detailed Description	11
5.1.2	Constructor & Destructor Documentation	11
5.1.2.1	AboutWindow	11
5.1.3	Member Function Documentation	11
5.1.3.1	SetupHelpWindow	11

5.1.3.2	SetupWindowType	12
5.2	InvertedPendulumTransporter.App Class Reference	12
5.2.1	Detailed Description	12
5.3	InvertedPendulumTransporter.Models.Cart Class Reference	12
5.3.1	Detailed Description	13
5.3.2	Constructor & Destructor Documentation	13
5.3.2.1	Cart	13
5.3.3	Member Function Documentation	13
5.3.3.1	Initialize	13
5.3.3.2	SetupHighLevelGraphics	13
5.3.3.3	SetupLowLevelGraphics	13
5.3.3.4	UpdateState	14
5.3.4	Member Data Documentation	15
5.3.4.1	platformHeightFactor	15
5.3.4.2	wheelRadius	15
5.4	InvertedPendulumTransporter.Windows.CreateTrajectoryWindow Class Reference	15
5.4.1	Detailed Description	15
5.4.2	Constructor & Destructor Documentation	15
5.4.2.1	CreateTrajectoryWindow	15
5.4.3	Property Documentation	16
5.4.3.1	TrajectoryLoaded	16
5.4.3.2	TrajectoryPoints	16
5.5	InvertedPendulumTransporterPhysics.Controllers.GameController Class Reference	16
5.5.1	Detailed Description	16
5.5.2	Constructor & Destructor Documentation	17
5.5.2.1	GameController	17
5.5.3	Member Function Documentation	18
5.5.3.1	HandleKey	18
5.5.3.2	Reset	18
5.6	InvertedPendulumTransporter.Models.ICart Interface Reference	18
5.6.1	Detailed Description	18
5.6.2	Property Documentation	19
5.6.2.1	PlatformSize	19
5.7	InvertedPendulumTransporterPhysics.Controllers.IController Interface Reference	19
5.7.1	Detailed Description	19
5.7.2	Member Function Documentation	19
5.7.2.1	Reset	19
5.8	InvertedPendulumTransporterPhysics.Controllers.IGameController Interface Reference	19
5.8.1	Detailed Description	20
5.8.2	Member Function Documentation	20

5.8.2.1	HandleKey	20
5.8.3	Property Documentation	20
5.8.3.1	GameEnabled	20
5.8.3.2	GamePlaying	20
5.8.3.3	UserAngleX	21
5.8.3.4	UserAngleY	21
5.9	InvertedPendulumTransporter.Models.IModel Interface Reference	21
5.9.1	Detailed Description	21
5.9.2	Member Function Documentation	21
5.9.2.1	Initialize	21
5.9.2.2	SetupHighLevelGraphics	22
5.9.2.3	SetupLowLevelGraphics	22
5.9.2.4	UpdateState	22
5.9.3	Property Documentation	22
5.9.3.1	Model	22
5.10	InvertedPendulumTransporterPhysics.Solvers.InterferedSystemODESolverFunctionStrategy Class Reference	22
5.10.1	Detailed Description	23
5.10.2	Member Function Documentation	23
5.10.2.1	ODESolverFunction	23
5.11	InvertedPendulumTransporterPhysics.Solvers.IODESolver Interface Reference	23
5.11.1	Detailed Description	23
5.11.2	Member Function Documentation	23
5.11.2.1	SetupStrategy	24
5.11.2.2	SolveODESystem	25
5.11.2.3	UpdateSystemParameters	25
5.12	InvertedPendulumTransporterPhysics.Solvers.IODESolverFunctionStrategy Interface Reference	25
5.12.1	Detailed Description	25
5.12.2	Member Function Documentation	26
5.12.2.1	ODESolverFunction	26
5.13	InvertedPendulumTransporter.Models.IPendulum Interface Reference	26
5.13.1	Detailed Description	26
5.13.2	Property Documentation	26
5.13.2.1	CartLinkPoint	27
5.13.2.2	MassLinkPoint	27
5.13.2.3	RodLength	27
5.14	InvertedPendulumTransporter.Models.ISimulationArea Interface Reference	27
5.14.1	Detailed Description	27
5.14.2	Property Documentation	27
5.14.2.1	Size	27

5.15 InvertedPendulumTransporterPhysics.Controllers.ITrajectoryController Interface Reference . . . . .	28
5.15.1 Detailed Description . . . . .	28
5.15.2 Member Function Documentation . . . . .	28
5.15.2.1 Clear . . . . .	28
5.15.2.2 GetTargetPosition . . . . .	29
5.15.2.3 GetTargetStartPosition . . . . .	30
5.15.2.4 LoadTrajectory . . . . .	30
5.15.2.5 SaveTrajectory . . . . .	30
5.15.2.6 SetAccuracy . . . . .	30
5.15.3 Property Documentation . . . . .	31
5.15.3.1 AverageDistance . . . . .	31
5.15.3.2 TrajectoryAchieved . . . . .	31
5.15.3.3 TrajectoryEnabled . . . . .	31
5.16 InvertedPendulumTransporterPhysics.Controllers.IVoltageController Interface Reference . . . . .	31
5.16.1 Detailed Description . . . . .	32
5.16.2 Member Function Documentation . . . . .	32
5.16.2.1 GetVoltage . . . . .	32
5.16.2.2 Reset . . . . .	32
5.16.2.3 SetControlError . . . . .	32
5.16.2.4 SetTime . . . . .	32
5.16.2.5 SetUserAngle . . . . .	32
5.16.3 Property Documentation . . . . .	33
5.16.3.1 ControlType . . . . .	33
5.17 InvertedPendulumTransporterPhysics.Controllers.IWindController Interface Reference . . . . .	33
5.17.1 Detailed Description . . . . .	34
5.17.2 Member Function Documentation . . . . .	34
5.17.2.1 GetXCoordWindPower . . . . .	34
5.17.2.2 GetYCoordWindPower . . . . .	34
5.17.2.3 GetZCoordWindPower . . . . .	34
5.17.2.4 UpdateWindForce . . . . .	34
5.17.3 Property Documentation . . . . .	34
5.17.3.1 DefaultWindPower . . . . .	34
5.17.3.2 MaxWindPower . . . . .	35
5.17.3.3 MinWindPower . . . . .	35
5.17.3.4 WindChangeSpeed . . . . .	35
5.17.3.5 WindPower . . . . .	35
5.17.3.6 WindType . . . . .	35
5.18 InvertedPendulumTransporter.Windows.MainWindow Class Reference . . . . .	35
5.18.1 Detailed Description . . . . .	36
5.18.2 Constructor & Destructor Documentation . . . . .	36

5.18.2.1	MainWindow	36
5.18.3	Property Documentation	37
5.18.3.1	CartMass	37
5.18.3.2	CartPositionX	37
5.18.3.3	CartPositionY	37
5.18.3.4	CartVelocityX	37
5.18.3.5	CartVelocityY	37
5.18.3.6	MaxAngle	37
5.18.3.7	MaxWindPower	37
5.18.3.8	MinAngle	37
5.18.3.9	MinWindPower	37
5.18.3.10	PendulumAngleX	37
5.18.3.11	PendulumAngleY	37
5.18.3.12	PendulumAngularVelocityX	37
5.18.3.13	PendulumAngularVelocityY	38
5.18.3.14	PendulumMass	38
5.18.3.15	RodLength	38
5.18.3.16	TimeDelta	38
5.18.3.17	WindChangeSpeed	38
5.18.3.18	WindPower	38
5.18.3.19	XCoordAngle	38
5.18.3.20	YCoordAngle	38
5.18.4	Event Documentation	38
5.18.4.1	PropertyChanged	38
5.19	InvertedPendulumTransporterPhysics.Solvers.ODESolver Class Reference	38
5.19.1	Detailed Description	39
5.19.2	Constructor & Destructor Documentation	39
5.19.2.1	ODESolver	39
5.19.3	Member Function Documentation	39
5.19.3.1	SetupStrategy	39
5.19.3.2	SolveODESystem	39
5.19.3.3	UpdateSystemParameters	40
5.20	InvertedPendulumTransporterPhysics.Common.OneDimensionalSystemState Class Reference	40
5.20.1	Detailed Description	41
5.20.2	Constructor & Destructor Documentation	41
5.20.2.1	OneDimensionalSystemState	41
5.20.2.2	OneDimensionalSystemState	41
5.20.3	Member Function Documentation	41
5.20.3.1	InitializeDefault	41
5.20.3.2	ToStateArray	41

5.20.4	Property Documentation	41
5.20.4.1	Angle	41
5.20.4.2	AngularVelocity	41
5.20.4.3	Position	41
5.20.4.4	Velocity	42
5.21	InvertedPendulumTransporter.Models.Pendulum Class Reference	42
5.21.1	Detailed Description	42
5.21.2	Constructor & Destructor Documentation	42
5.21.2.1	Pendulum	42
5.21.3	Member Function Documentation	43
5.21.3.1	Initialize	43
5.21.3.2	SetupHighLevelGraphics	43
5.21.3.3	SetupLowLevelGraphics	43
5.21.3.4	UpdateState	43
5.22	InvertedPendulumTransporter.Physics.Controllers.PIDCorrector Class Reference	43
5.22.1	Detailed Description	44
5.22.2	Constructor & Destructor Documentation	44
5.22.2.1	PIDCorrector	44
5.22.3	Member Function Documentation	44
5.22.3.1	CalculateAnglePIDCorrection	44
5.22.3.2	CalculateParallelPositionAnglePIDCorrection	44
5.22.3.3	CalculatePositionPIDCorrection	44
5.22.3.4	Reset	44
5.22.3.5	Reset	44
5.22.3.6	SetAngleError	45
5.22.3.7	SetPositionError	45
5.23	InvertedPendulumTransporter.Controls.PlotsControl Class Reference	45
5.23.1	Detailed Description	46
5.23.2	Constructor & Destructor Documentation	46
5.23.2.1	PlotsControl	46
5.23.3	Member Function Documentation	46
5.23.3.1	PassParameters	46
5.23.3.2	ResetPlots	46
5.23.3.3	UpdateAngleErrorPlots	46
5.23.3.4	UpdatePositionErrorPlots	47
5.23.3.5	UpdateVoltagePlots	47
5.23.4	Property Documentation	47
5.23.4.1	AngleErrorPointsX	47
5.23.4.2	AngleErrorPointsY	47
5.23.4.3	PositionErrorPointsX	47



5.23.4.4	PositionErrorPointsY	47
5.23.4.5	VoltagePointsX	47
5.23.4.6	VoltagePointsY	48
5.24	InvertedPendulumTransporter.Controls.SceneControl Class Reference	48
5.24.1	Detailed Description	49
5.24.2	Constructor & Destructor Documentation	49
5.24.2.1	SceneControl	49
5.24.3	Member Function Documentation	49
5.24.3.1	ClearTrajectory	49
5.24.3.2	ResetSimulation	49
5.24.3.3	SetupHighLevelGraphics	49
5.24.3.4	SetupLowLevelGraphics	49
5.24.3.5	ShowCartTrajectory	49
5.24.3.6	ShowPendulumTrajectory	50
5.24.3.7	ShowTargetTrajectory	50
5.24.3.8	UpdateCamera	50
5.24.3.9	UpdateFrame	50
5.24.3.10	UpdateState	50
5.24.3.11	UpdateTrajectory	50
5.24.3.12	UpdateWindDirection	51
5.24.4	Member Data Documentation	51
5.24.4.1	cart	51
5.24.4.2	pendulum	51
5.24.4.3	simulationArea	51
5.25	InvertedPendulumTransporter.Models.SimulationArea Class Reference	51
5.25.1	Detailed Description	52
5.25.2	Constructor & Destructor Documentation	52
5.25.2.1	SimulationArea	52
5.25.3	Member Function Documentation	52
5.25.3.1	Initialize	52
5.25.3.2	SetupHighLevelGraphics	52
5.25.3.3	SetupLowLevelGraphics	52
5.25.3.4	UpdateState	52
5.26	InvertedPendulumTransporterPhysics.Common.SolverParameters Class Reference	53
5.26.1	Detailed Description	53
5.26.2	Constructor & Destructor Documentation	53
5.26.2.1	SolverParameters	53
5.26.3	Member Data Documentation	54
5.26.3.1	G	54
5.26.4	Property Documentation	54

5.26.4.1	<a href="#">CartMass</a>	54
5.26.4.2	<a href="#">Gamma1</a>	54
5.26.4.3	<a href="#">Gamma2</a>	54
5.26.4.4	<a href="#">HorizontalWindForce</a>	54
5.26.4.5	<a href="#">PendulumLength</a>	54
5.26.4.6	<a href="#">PendulumMass</a>	54
5.26.4.7	<a href="#">VerticalWindForce</a>	54
5.26.4.8	<a href="#">Voltage</a>	54
5.27	<a href="#">InvertedPendulumTransporterPhysics.Solvers.StandardSystemODESolverFunctionStrategy Class Reference</a>	54
5.27.1	<a href="#">Detailed Description</a>	55
5.27.2	<a href="#">Member Function Documentation</a>	55
5.27.2.1	<a href="#">ODESolverFunction</a>	55
5.28	<a href="#">InvertedPendulumTransporterPhysics.Common.SystemState Class Reference</a>	55
5.28.1	<a href="#">Detailed Description</a>	56
5.28.2	<a href="#">Constructor &amp; Destructor Documentation</a>	56
5.28.2.1	<a href="#">SystemState</a>	56
5.28.3	<a href="#">Member Function Documentation</a>	57
5.28.3.1	<a href="#">GetSystemPosition</a>	57
5.28.3.2	<a href="#">Reset</a>	57
5.28.3.3	<a href="#">ResetSystemParameters</a>	57
5.28.3.4	<a href="#">ResetTimer</a>	57
5.28.3.5	<a href="#">ToTimeArray</a>	57
5.28.3.6	<a href="#">UpdateSystemStateX</a>	57
5.28.3.7	<a href="#">UpdateSystemStateY</a>	57
5.28.3.8	<a href="#">UpdateTimer</a>	58
5.28.4	<a href="#">Member Data Documentation</a>	58
5.28.4.1	<a href="#">DefaultTimeDelta</a>	58
5.28.5	<a href="#">Property Documentation</a>	58
5.28.5.1	<a href="#">LastStateX</a>	58
5.28.5.2	<a href="#">LastStateY</a>	58
5.28.5.3	<a href="#">MaxAngle</a>	58
5.28.5.4	<a href="#">MinAngle</a>	58
5.28.5.5	<a href="#">SolverParameters</a>	58
5.28.5.6	<a href="#">StateX</a>	58
5.28.5.7	<a href="#">StateY</a>	58
5.28.5.8	<a href="#">Time</a>	58
5.28.5.9	<a href="#">TimeDelta</a>	59
5.29	<a href="#">InvertedPendulumTransporterPhysics.Controllers.TrajectoryController Class Reference</a>	59
5.29.1	<a href="#">Detailed Description</a>	60

5.29.2	Constructor & Destructor Documentation	60
5.29.2.1	TrajectoryController	60
5.29.3	Member Function Documentation	60
5.29.3.1	Clear	60
5.29.3.2	GetTargetApproximatePosition	60
5.29.3.3	GetTargetPosition	60
5.29.3.4	GetTargetSmoothPosition	60
5.29.3.5	GetTargetStartPosition	61
5.29.3.6	LoadTrajectory	61
5.29.3.7	Reset	61
5.29.3.8	SaveTrajectory	61
5.29.3.9	SetAccuracy	62
5.30	InvertedPendulumTransporterPhysics.Controllers.VoltageController Class Reference	63
5.30.1	Detailed Description	64
5.30.2	Constructor & Destructor Documentation	64
5.30.2.1	VoltageController	64
5.30.3	Member Function Documentation	64
5.30.3.1	GetVoltage	64
5.30.3.2	Reset	64
5.30.3.3	Reset	64
5.30.3.4	SetControlError	64
5.30.3.5	SetTime	64
5.30.3.6	SetUserAngle	65
5.31	InvertedPendulumTransporterPhysics.Controllers.WindController Class Reference	65
5.31.1	Detailed Description	66
5.31.2	Constructor & Destructor Documentation	66
5.31.2.1	WindController	66
5.31.3	Member Function Documentation	66
5.31.3.1	GetXCoordWindPower	66
5.31.3.2	GetYCoordWindPower	66
5.31.3.3	GetZCoordWindPower	66
5.31.3.4	Reset	67
5.31.3.5	UpdateWindForce	67
5.31.4	Member Data Documentation	67
5.31.4.1	DefaultWindChangeSpeed	67
5.31.4.2	DefaultWindType	67
	<b>Index</b>	<b>69</b>



# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

<a href="#">InvertedPendulumTransporter</a>	7
<a href="#">InvertedPendulumTransporter.Controls</a>	7
<a href="#">InvertedPendulumTransporter.Models</a>	7
<a href="#">InvertedPendulumTransporter.Properties</a>	8
<a href="#">InvertedPendulumTransporter.Windows</a>	8
<a href="#">InvertedPendulumTransporterPhysics</a>	8
<a href="#">InvertedPendulumTransporterPhysics.Common</a>	9
<a href="#">InvertedPendulumTransporterPhysics.Controllers</a>	9
<a href="#">InvertedPendulumTransporterPhysics.Solvers</a>	10



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Application	
InvertedPendulumTransporter.App	12
InvertedPendulumTransporterPhysics.Controllers.IController	19
InvertedPendulumTransporterPhysics.Controllers.IGameController	19
InvertedPendulumTransporterPhysics.Controllers.GameController	16
InvertedPendulumTransporterPhysics.Controllers.ITrajectoryController	28
InvertedPendulumTransporterPhysics.Controllers.TrajectoryController	59
InvertedPendulumTransporterPhysics.Controllers.IVoltageController	31
InvertedPendulumTransporterPhysics.Controllers.VoltageController	63
InvertedPendulumTransporterPhysics.Controllers.IWindController	33
InvertedPendulumTransporterPhysics.Controllers.WindController	65
InvertedPendulumTransporter.Models.IModel	21
InvertedPendulumTransporter.Models.ICart	18
InvertedPendulumTransporter.Models.Cart	12
InvertedPendulumTransporter.Models.IPendulum	26
InvertedPendulumTransporter.Models.Pendulum	42
InvertedPendulumTransporter.Models.ISimulationArea	27
InvertedPendulumTransporter.Models.SimulationArea	51
INotifyPropertyChanged	
InvertedPendulumTransporter.Windows.MainWindow	35
InvertedPendulumTransporterPhysics.Solvers.IODESolver	23
InvertedPendulumTransporterPhysics.Solvers.ODESolver	38
InvertedPendulumTransporterPhysics.Solvers.IODESolverFunctionStrategy	25
InvertedPendulumTransporterPhysics.Solvers.InterferedSystemODESolverFunctionStrategy	22
InvertedPendulumTransporterPhysics.Solvers.StandardSystemODESolverFunctionStrategy	54
InvertedPendulumTransporterPhysics.Common.OneDimensionalSystemState	40
InvertedPendulumTransporterPhysics.Controllers.PIDCorrector	43
InvertedPendulumTransporterPhysics.Common.SolverParameters	53
InvertedPendulumTransporterPhysics.Common.SystemState	55
UserControl	
InvertedPendulumTransporter.Controls.PlotsControl	45
InvertedPendulumTransporter.Controls.SceneControl	48
Window	
InvertedPendulumTransporter.Windows.AboutWindow	11
InvertedPendulumTransporter.Windows.CreateTrajectoryWindow	15
InvertedPendulumTransporter.Windows.MainWindow	35





## Chapter 3

# Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">InvertedPendulumTransporter.Windows.AboutWindow</a>	
Interaction logic for AboutWindow.xaml . . . . .	11
<a href="#">InvertedPendulumTransporter.App</a>	
Interaction logic for App.xaml . . . . .	12
<a href="#">InvertedPendulumTransporter.Models.Cart</a>	
Cart model class . . . . .	12
<a href="#">InvertedPendulumTransporter.Windows.CreateTrajectoryWindow</a>	
Interaction logic for CreateTrajectoryWindow.xaml . . . . .	15
<a href="#">InvertedPendulumTransporterPhysics.Controllers.GameController</a>	
Controller for game mode . . . . .	16
<a href="#">InvertedPendulumTransporter.Models.ICart</a>	
Interface dedicated to cart model . . . . .	18
<a href="#">InvertedPendulumTransporterPhysics.Controllers.IController</a>	
Interface dedicated to all physics controllers . . . . .	19
<a href="#">InvertedPendulumTransporterPhysics.Controllers.IGameController</a>	
Interface dedicated to game controller . . . . .	19
<a href="#">InvertedPendulumTransporter.Models.IModel</a>	
Interface dedicated to all visualization models . . . . .	21
<a href="#">InvertedPendulumTransporterPhysics.Solvers.InterferedSystemODESolverFunctionStrategy</a>	
Strategy for dynamics system with interferences . . . . .	22
<a href="#">InvertedPendulumTransporterPhysics.Solvers.IODESolver</a>	
Interface dedicated to ODE solver . . . . .	23
<a href="#">InvertedPendulumTransporterPhysics.Solvers.IODESolverFunctionStrategy</a>	
Interface dedicated to ODE solving function strategies . . . . .	25
<a href="#">InvertedPendulumTransporter.Models.IPendulum</a>	
Interface dedicated to pendulum model . . . . .	26
<a href="#">InvertedPendulumTransporter.Models.ISimulationArea</a>	
Interface dedicated to simulation area model . . . . .	27
<a href="#">InvertedPendulumTransporterPhysics.Controllers.ITrajectoryController</a>	
Interface dedicated to trajectory controller . . . . .	28
<a href="#">InvertedPendulumTransporterPhysics.Controllers.IVoltageController</a>	
Interface dedicated to voltage controller . . . . .	31
<a href="#">InvertedPendulumTransporterPhysics.Controllers.IWindController</a>	
Interface dedicated to wind controller . . . . .	33
<a href="#">InvertedPendulumTransporter.Windows.MainWindow</a>	
Interaction logic for MainWindow.xaml [application management] . . . . .	35
<a href="#">InvertedPendulumTransporterPhysics.Solvers.ODESolver</a>	
ODE solver class . . . . .	38

<a href="#">InvertedPendulumTransporterPhysics.Common.OneDimensionalSystemState</a>	
Container for subsystem state . . . . .	40
<a href="#">InvertedPendulumTransporter.Models.Pendulum</a>	
Pendulum model class . . . . .	42
<a href="#">InvertedPendulumTransporterPhysics.Controllers.PIDCorrector</a>	
Helping class for voltage controlling dedicated to PID corrector . . . . .	43
<a href="#">InvertedPendulumTransporter.Controls.PlotsControl</a>	
Interaction logic for PlotsControl.xaml . . . . .	45
<a href="#">InvertedPendulumTransporter.Controls.SceneControl</a>	
Interaction logic for Scene.xaml . . . . .	48
<a href="#">InvertedPendulumTransporter.Models.SimulationArea</a>	
Simulation area model class . . . . .	51
<a href="#">InvertedPendulumTransporterPhysics.Common.SolverParameters</a>	
Container for solver parameters . . . . .	53
<a href="#">InvertedPendulumTransporterPhysics.Solvers.StandardSystemODESolverFunctionStrategy</a>	
Strategy for base dynamics system . . . . .	54
<a href="#">InvertedPendulumTransporterPhysics.Common.SystemState</a>	
Container for the whole system state . . . . .	55
<a href="#">InvertedPendulumTransporterPhysics.Controllers.TrajectoryController</a>	
Controller for trajectory tracking . . . . .	59
<a href="#">InvertedPendulumTransporterPhysics.Controllers.VoltageController</a>	
Controller for motor voltage . . . . .	63
<a href="#">InvertedPendulumTransporterPhysics.Controllers.WindController</a>	
Controller for wind power . . . . .	65

## Chapter 4

# Namespace Documentation

### 4.1 Package InvertedPendulumTransporter

#### Namespaces

- package [Controls](#)
- package [Models](#)
- package [Properties](#)
- package [Windows](#)

#### Classes

- class [App](#)  
*Interaction logic for App.xaml*

### 4.2 Package InvertedPendulumTransporter.Controls

#### Classes

- class [PlotsControl](#)  
*Interaction logic for PlotsControl.xaml*
- class [SceneControl](#)  
*Interaction logic for Scene.xaml*

### 4.3 Package InvertedPendulumTransporter.Models

#### Classes

- class [Cart](#)  
*[Cart](#) model class*
- interface [ICart](#)  
*Interface dedicated to cart model*
- interface [IModel](#)  
*Interface dedicated to all visualization models*
- interface [IPendulum](#)  
*Interface dedicated to pendulum model*

- interface [ISimulationArea](#)  
*Interface dedicated to simulation area model*
- class [Pendulum](#)  
*Pendulum model class*
- class [SimulationArea](#)  
*Simulation area model class*

## 4.4 Package InvertedPendulumTransporter.Properties

### Classes

- class **Resources**  
*A strongly-typed resource class, for looking up localized strings, etc.*
- class **Settings**

## 4.5 Package InvertedPendulumTransporter.Windows

### Classes

- class [AboutWindow](#)  
*Interaction logic for AboutWindow.xaml*
- class [CreateTrajectoryWindow](#)  
*Interaction logic for CreateTrajectoryWindow.xaml*
- class [MainWindow](#)  
*Interaction logic for MainWindow.xaml [application management]*

### Enumerations

- enum [WindowType](#) {  
**Application, SystemMechanics, Author, HelpMenuMode,**  
**HelpMenuOptions, HelpMenuAbout, HelpAnimationPanel, HelpSystemParameters,**  
**HelpWindParameters, HelpSystemStateInfo, HelpSimulationScene, HelpPlots** }  
*Enumeration for window types*

### 4.5.1 Enumeration Type Documentation

#### 4.5.1.1 enum InvertedPendulumTransporter.Windows.WindowType

Enumeration for window types

## 4.6 Package InvertedPendulumTransporterPhysics

### Namespaces

- package [Common](#)
- package [Controllers](#)
- package [Solvers](#)

## 4.7 Package InvertedPendulumTransporterPhysics.Common

### Classes

- class [OneDimensionalSystemState](#)  
*Container for subsystem state*
- class [SolverParameters](#)  
*Container for solver parameters*
- class [SystemState](#)  
*Container for the whole system state*

## 4.8 Package InvertedPendulumTransporterPhysics.Controllers

### Classes

- class [GameController](#)  
*Controller for game mode*
- interface [IController](#)  
*Interface dedicated to all physics controllers*
- interface [IGameController](#)  
*Interface dedicated to game controller*
- interface [ITrajectoryController](#)  
*Interface dedicated to trajectory controller*
- interface [IVoltageController](#)  
*Interface dedicated to voltage controller*
- interface [IWindController](#)  
*Interface dedicated to wind controller*
- class [PIDCorrector](#)  
*Helping class for voltage controlling dedicated to PID corrector*
- class [TrajectoryController](#)  
*Controller for trajectory tracking*
- class [VoltageController](#)  
*Controller for motor voltage*
- class [WindController](#)  
*Controller for wind power*

### Enumerations

- enum [AccuracyType](#) { **Ultra**, **High**, **Medium**, **Low** }  
*Enumeration for tracking accuracy types*
- enum [ControlType](#) { **Random**, **Sinusoidal**, **None**, **PID**, **DoublePIDCascade**, **DoublePIDParallel**, **DoublePDPParallel** }  
*Enumeration for control types*
- enum [WindType](#) { **RandomPeak**, **RandomSwitch**, **RandomSmooth** }  
*Enumeration for wind generation types*

### 4.8.1 Enumeration Type Documentation

#### 4.8.1.1 enum InvertedPendulumTransporterPhysics.Controllers.AccuracyType

Enumeration for tracking accuracy types

#### 4.8.1.2 enum InvertedPendulumTransporterPhysics.Controllers.ControlType

Enumeration for control types

#### 4.8.1.3 enum InvertedPendulumTransporterPhysics.Controllers.WindType

Enumeration for wind generation types

## 4.9 Package InvertedPendulumTransporterPhysics.Solvers

### Classes

- class [InterferedSystemODESolverFunctionStrategy](#)  
*Strategy for dynamics system with interferences*
- interface [IODESolver](#)  
*Interface dedicated to ODE solver*
- interface [IODESolverFunctionStrategy](#)  
*Interface dedicated to ODE solving function strategies*
- class [ODESolver](#)  
*ODE solver class*
- class [StandardSystemODESolverFunctionStrategy](#)  
*Strategy for base dynamics system*

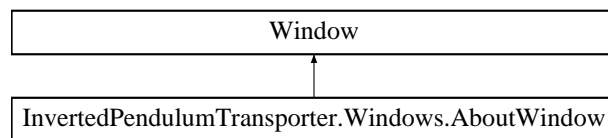
## Chapter 5

# Class Documentation

### 5.1 InvertedPendulumTransporter.Windows.AboutWindow Class Reference

Interaction logic for AboutWindow.xaml

Inheritance diagram for InvertedPendulumTransporter.Windows.AboutWindow:



#### Public Member Functions

- [AboutWindow](#) ()  
*Class constructor*
- void [SetupWindowType](#) ([WindowType](#) type)  
*Setup window type*
- void [SetupHelpWindow](#) (bool finish)  
*Mark that actual window is dedicated to help*

#### 5.1.1 Detailed Description

Interaction logic for AboutWindow.xaml

#### 5.1.2 Constructor & Destructor Documentation

##### 5.1.2.1 InvertedPendulumTransporter.Windows.AboutWindow.AboutWindow ( ) [inline]

Class constructor

#### 5.1.3 Member Function Documentation

##### 5.1.3.1 void InvertedPendulumTransporter.Windows.AboutWindow.SetupHelpWindow ( bool *finish* ) [inline]

Mark that actual window is dedicated to help

## Parameters

<i>finish</i>	Is window the last in help
---------------	----------------------------

5.1.3.2 void InvertedPendulumTransporter.Windows.AboutWindow.SetupWindowType ( WindowType *type* ) [inline]

Setup window type

## Parameters

<i>type</i>	Window type
-------------	-------------

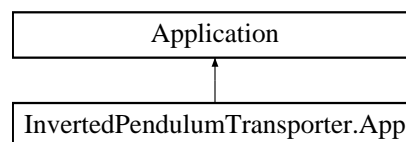
The documentation for this class was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/Simulation/Windows/AboutWindow.xaml.cs

## 5.2 InvertedPendulumTransporter.App Class Reference

Interaction logic for App.xaml

Inheritance diagram for InvertedPendulumTransporter.App:



### 5.2.1 Detailed Description

Interaction logic for App.xaml

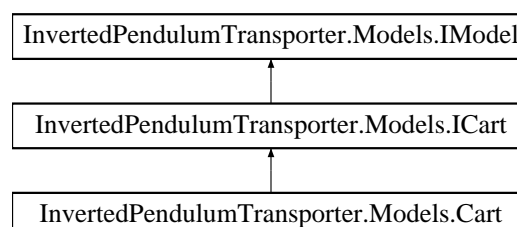
The documentation for this class was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/Simulation/App.xaml.cs

## 5.3 InvertedPendulumTransporter.Models.Cart Class Reference

[Cart](#) model class

Inheritance diagram for InvertedPendulumTransporter.Models.Cart:



### Public Member Functions

- [Cart](#) ()



*Class constructor*

- void [Initialize](#) ()

*Initialization method*

- void [UpdateState](#) ([SystemState](#) systemState)

*Uptate model visual*

- void [SetupHighLevelGraphics](#) ()

*Seup high level graphics for model*

- void [SetupLowLevelGraphics](#) ()

*Setup low level graphics for model*

## Public Attributes

- double [platformHeightFactor](#) = 0.1

*Factor between walls height and platform size*

- double [wheelRadius](#) = 0.5

*Wheel radius*

## Properties

- [ModelVisual3D](#) **Model** [get]
- double **PlatformSize** [get]

### 5.3.1 Detailed Description

[Cart](#) model class

### 5.3.2 Constructor & Destructor Documentation

#### 5.3.2.1 InvertedPendulumTransporter.Models.Cart.Cart ( ) [inline]

Class constructor

### 5.3.3 Member Function Documentation

#### 5.3.3.1 void InvertedPendulumTransporter.Models.Cart.Initialize ( ) [inline]

Initialization method

Implements [InvertedPendulumTransporter.Models.IModel](#).

#### 5.3.3.2 void InvertedPendulumTransporter.Models.Cart.SetupHighLevelGraphics ( ) [inline]

Seup high level graphics for model

Implements [InvertedPendulumTransporter.Models.IModel](#).

#### 5.3.3.3 void InvertedPendulumTransporter.Models.Cart.SetupLowLevelGraphics ( ) [inline]

Setup low level graphics for model

Implements [InvertedPendulumTransporter.Models.IModel](#).

5.3.3.4 void InvertedPendulumTransporter.Models.Cart.UpdateState ( **SystemState** *systemState* ) [inline]

Uptate model visual

## Parameters

<i>systemState</i>	Actual system state
--------------------	---------------------

Implements [InvertedPendulumTransporter.Models.IModel](#).

### 5.3.4 Member Data Documentation

#### 5.3.4.1 double InvertedPendulumTransporter.Models.Cart.platformHeightFactor = 0.1

Factor between walls height and platform size

#### 5.3.4.2 double InvertedPendulumTransporter.Models.Cart.wheelRadius = 0.5

Wheel radius

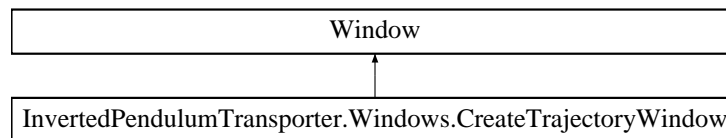
The documentation for this class was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/Simulation/Models/Cart.cs

## 5.4 InvertedPendulumTransporter.Windows.CreateTrajectoryWindow Class Reference

Interaction logic for CreateTrajectoryWindow.xaml

Inheritance diagram for InvertedPendulumTransporter.Windows.CreateTrajectoryWindow:



### Public Member Functions

- [CreateTrajectoryWindow](#) ()  
*Class constructor*

### Properties

- List< Point3D > [TrajectoryPoints](#) [get]  
*Result trajectory points*
- bool [TrajectoryLoaded](#) [get]  
*Information about trajectory creation succes*

#### 5.4.1 Detailed Description

Interaction logic for CreateTrajectoryWindow.xaml

#### 5.4.2 Constructor & Destructor Documentation

##### 5.4.2.1 InvertedPendulumTransporter.Windows.CreateTrajectoryWindow.CreateTrajectoryWindow ( ) [inline]

Class constructor

### 5.4.3 Property Documentation

5.4.3.1 `bool InvertedPendulumTransporter.Windows.CreateTrajectoryWindow.TrajectoryLoaded` [get]

Information about trajectory creation succes

5.4.3.2 `List<Point3D> InvertedPendulumTransporter.Windows.CreateTrajectoryWindow.TrajectoryPoints` [get]

Result trajectory points

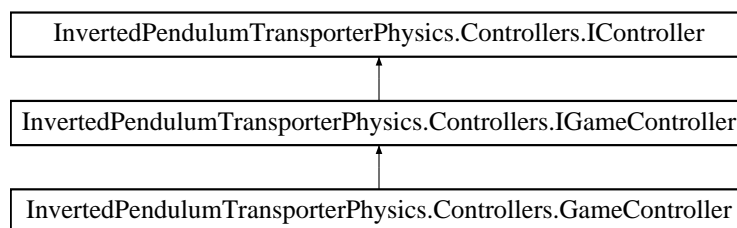
The documentation for this class was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/Simulation/Windows/CreateTrajectoryWindow.xaml.cs

## 5.5 InvertedPendulumTransporterPhysics.Controllers.GameController Class Reference

Controller for game mode

Inheritance diagram for InvertedPendulumTransporterPhysics.Controllers.GameController:



### Public Member Functions

- [GameController](#) (Button up, Button down, Button left, Button right)

*Class constructor*

- void [Reset](#) ()

*Reset controller to default state*

- void [HandleKey](#) (Key key)

*User actions handlers*

### Properties

- double **UserAngleX** [get]
- double **UserAngleY** [get]
- bool **GameEnabled** [get, set]
- bool **GamePlaying** [get, set]

#### 5.5.1 Detailed Description

Controller for game mode

## 5.5.2 Constructor & Destructor Documentation

### 5.5.2.1 InvertedPendulumTransporterPhysics.Controllers.GameController.GameController ( Button *up*, Button *down*, Button *left*, Button *right* ) `[inline]`

Class constructor

## Parameters

<i>up</i>	Up key binding
<i>down</i>	Down key binding
<i>left</i>	Left key binding
<i>right</i>	Right key binding

## 5.5.3 Member Function Documentation

5.5.3.1 `void InvertedPendulumTransporterPhysics.Controllers.GameController.HandleKey ( Key key ) [inline]`

User actions handlers

## Parameters

<i>key</i>	Keyboard key
------------	--------------

Implements [InvertedPendulumTransporterPhysics.Controllers.IGameController](#).

5.5.3.2 `void InvertedPendulumTransporterPhysics.Controllers.GameController.Reset ( ) [inline]`

Reset controller to default state

Implements [InvertedPendulumTransporterPhysics.Controllers.IController](#).

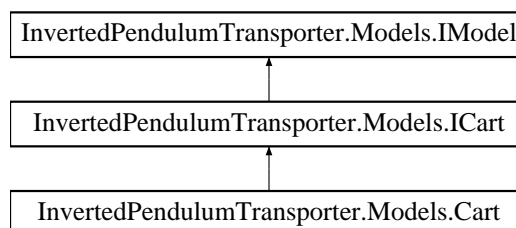
The documentation for this class was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/PhysicsLibrary/Controllers/GameController.cs

## 5.6 InvertedPendulumTransporter.Models.ICart Interface Reference

Interface dedicated to cart model

Inheritance diagram for InvertedPendulumTransporter.Models.ICart:



## Properties

- double [PlatformSize](#) [get]  
*Cart platform size*

## Additional Inherited Members

## 5.6.1 Detailed Description

Interface dedicated to cart model

## 5.6.2 Property Documentation

### 5.6.2.1 double InvertedPendulumTransporter.Models.ICart.PlatformSize [get]

[Cart](#) platform size

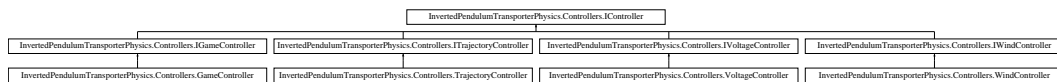
The documentation for this interface was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/Simulation/Models/ICart.cs

## 5.7 InvertedPendulumTransporterPhysics.Controllers.IController Interface Reference

Interface dedicated to all physics controllers

Inheritance diagram for InvertedPendulumTransporterPhysics.Controllers.IController:



## Public Member Functions

- void [Reset](#) ()  
*Reset controller to default state*

### 5.7.1 Detailed Description

Interface dedicated to all physics controllers

### 5.7.2 Member Function Documentation

#### 5.7.2.1 void InvertedPendulumTransporterPhysics.Controllers.IController.Reset ( )

Reset controller to default state

Implemented in [InvertedPendulumTransporterPhysics.Controllers.WindController](#), [InvertedPendulumTransporterPhysics.Controllers.TrajectoryController](#), [InvertedPendulumTransporterPhysics.Controllers.VoltageController](#), and [InvertedPendulumTransporterPhysics.Controllers.GameController](#).

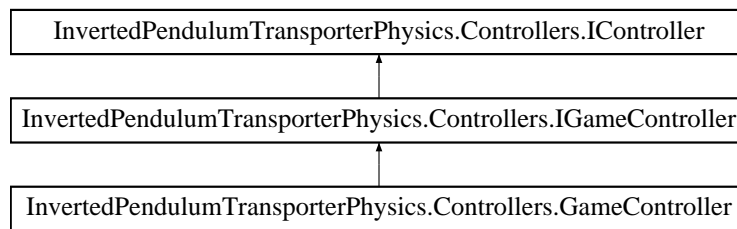
The documentation for this interface was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/PhysicsLibrary/Controllers/IController.cs

## 5.8 InvertedPendulumTransporterPhysics.Controllers.IGameController Interface Reference

Interface dedicated to game controller

Inheritance diagram for InvertedPendulumTransporterPhysics.Controllers.IGameController:



## Public Member Functions

- void [HandleKey](#) (Key key)

*User actions handlers*

## Properties

- double [UserAngleX](#) [get]  
*Target pendulum angle generated by user in X-coordintate*
- double [UserAngleY](#) [get]  
*Target pendulum angle generated by user in Y-coordintate*
- bool [GameEnabled](#) [get, set]  
*Game mode enablement*
- bool [GamePlaying](#) [get, set]  
*Game progress enablement*

### 5.8.1 Detailed Description

Interface dedicated to game controller

### 5.8.2 Member Function Documentation

#### 5.8.2.1 void InvertedPendulumTransporterPhysics.Controllers.IGameController.HandleKey ( Key key )

User actions handlers

Parameters

<i>key</i>	Keyboard key
------------	--------------

Implemented in [InvertedPendulumTransporterPhysics.Controllers.GameController](#).

### 5.8.3 Property Documentation

#### 5.8.3.1 bool InvertedPendulumTransporterPhysics.Controllers.IGameController.GameEnabled [get], [set]

Game mode enablement

#### 5.8.3.2 bool InvertedPendulumTransporterPhysics.Controllers.IGameController.GamePlaying [get], [set]

Game progress enablement



5.8.3.3 `double InvertedPendulumTransporterPhysics.Controllers.IGameController.UserAngleX` `[get]`

Target pendulum angle generated by user in X-coordintate

5.8.3.4 `double InvertedPendulumTransporterPhysics.Controllers.IGameController.UserAngleY` `[get]`

Target pendulum angle generated by user in Y-coordintate

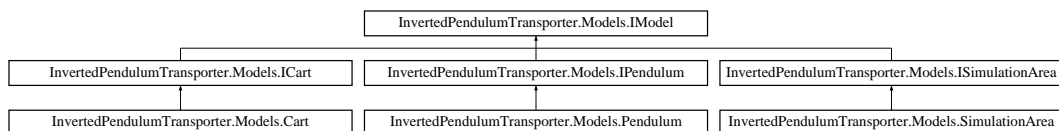
The documentation for this interface was generated from the following file:

- `C:/Users/Arkady/Downloads/Murzyński Backup/BIB/PhysicsLibrary/Controllers/IGameController.cs`

## 5.9 InvertedPendulumTransporter.Models.IModel Interface Reference

Interface dedicated to all visualization models

Inheritance diagram for InvertedPendulumTransporter.Models.IModel:



### Public Member Functions

- `void Initialize ()`  
*Initialization method*
- `void UpdateState (SystemState systemState)`  
*Uptate model visual*
- `void SetupHighLevelGraphics ()`  
*Seup high level graphics for model*
- `void SetupLowLevelGraphics ()`  
*Setup low level graphics for model*

### Properties

- `ModelVisual3D Model` `[get]`  
*Reference to model*

### 5.9.1 Detailed Description

Interface dedicated to all visualization models

### 5.9.2 Member Function Documentation

5.9.2.1 `void InvertedPendulumTransporter.Models.IModel.Initialize ( )`

Initialization method

Implemented in `InvertedPendulumTransporter.Models.Cart`, `InvertedPendulumTransporter.Models.Pendulum`, and `InvertedPendulumTransporter.Models.SimulationArea`.

### 5.9.2.2 void InvertedPendulumTransporter.Models.IModel.SetupHighLevelGraphics ( )

Seup high level graphics for model

Implemented in [InvertedPendulumTransporter.Models.Pendulum](#), [InvertedPendulumTransporter.Models.Cart](#), and [InvertedPendulumTransporter.Models.SimulationArea](#).

### 5.9.2.3 void InvertedPendulumTransporter.Models.IModel.SetupLowLevelGraphics ( )

Setup low level graphics for model

Implemented in [InvertedPendulumTransporter.Models.Pendulum](#), [InvertedPendulumTransporter.Models.Cart](#), and [InvertedPendulumTransporter.Models.SimulationArea](#).

### 5.9.2.4 void InvertedPendulumTransporter.Models.IModel.UpdateState ( SystemState systemState )

Uptate model visual

Parameters

<i>systemState</i>	Actual system state
--------------------	---------------------

Implemented in [InvertedPendulumTransporter.Models.Cart](#), [InvertedPendulumTransporter.Models.Pendulum](#), and [InvertedPendulumTransporter.Models.SimulationArea](#).

## 5.9.3 Property Documentation

### 5.9.3.1 ModelVisual3D InvertedPendulumTransporter.Models.IModel.Model [get]

Reference to model

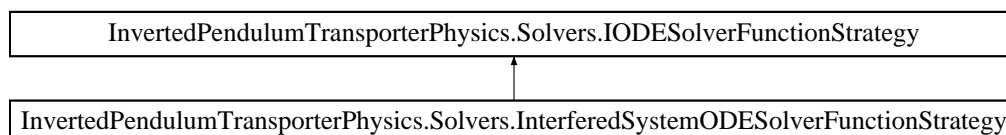
The documentation for this interface was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/Simulation/Models/IModel.cs

## 5.10 InvertedPendulumTransporterPhysics.Solvers.InterferedSystemODESolverFunction↔ Strategy Class Reference

Strategy for dynamics system with interferences

Inheritance diagram for InvertedPendulumTransporterPhysics.Solvers.InterferedSystemODESolverFunction↔ Strategy:



## Public Member Functions

- void [ODESolverFunction](#) (double[] y, double x, double[] dy, object obj)

*Function of solving the state-space equations in differential form*

### 5.10.1 Detailed Description

Strategy for dynamics system with interferences

### 5.10.2 Member Function Documentation

5.10.2.1 `void InvertedPendulumTransporterPhysics.Solvers.InterferedSystemODESolverFunctionStrategy.ODESolverFunction (double[] y, double x, double[] dy, object obj) [inline]`

Function of solving the state-space equations in differential form

Parameters

<i>y</i>	System state array
<i>x</i>	Time stamps array
<i>dy</i>	System state derivatives array
<i>obj</i>	Solver parameters

Implements [InvertedPendulumTransporterPhysics.Solvers.IODESolverFunctionStrategy](#).

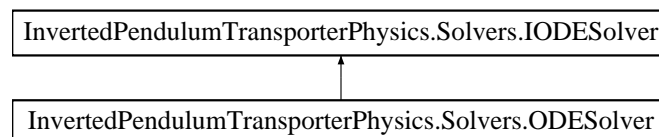
The documentation for this class was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/PhysicsLibrary/Solvers/InterferedSystemODESolver↔  
FunctionStrategy.cs

## 5.11 InvertedPendulumTransporterPhysics.Solvers.IODESolver Interface Reference

Interface dedicated to ODE solver

Inheritance diagram for InvertedPendulumTransporterPhysics.Solvers.IODESolver:



### Public Member Functions

- void [UpdateSystemParameters](#) ([SolverParameters](#) parameters)  
*Update solver parameters*
- [OneDimensionalSystemState SolveODESystem](#) (double[] *x*, double[] *y*)  
*Solve ordinary differential equation (ODE)*
- void [SetupStrategy](#) ([IODESolverFunctionStrategy](#) strategy)  
*Setup solving strategy*

### 5.11.1 Detailed Description

Interface dedicated to ODE solver

### 5.11.2 Member Function Documentation

5.11.2.1 void InvertedPendulumTransporterPhysics.Solvers.IODESolver.SetupStrategy ( IODESolverFunctionStrategy *strategy* )

Setup solving strategy

## Parameters

<i>strategy</i>	Strategy of solving equations
-----------------	-------------------------------

Implemented in [InvertedPendulumTransporterPhysics.Solvers.ODESolver](#).

### 5.11.2.2 OneDimensionalSystemState InvertedPendulumTransporterPhysics.Solvers.IODESolver.SolveODESystem ( double[] x, double[] y )

Solve ordinary differential equation (ODE)

## Parameters

<i>x</i>	System state array
<i>y</i>	Time stamps array

## Returns

New system state

Implemented in [InvertedPendulumTransporterPhysics.Solvers.ODESolver](#).

### 5.11.2.3 void InvertedPendulumTransporterPhysics.Solvers.IODESolver.UpdateSystemParameters ( SolverParameters parameters )

Update solver parameters

## Parameters

<i>parameters</i>	New solver parameters
-------------------	-----------------------

Implemented in [InvertedPendulumTransporterPhysics.Solvers.ODESolver](#).

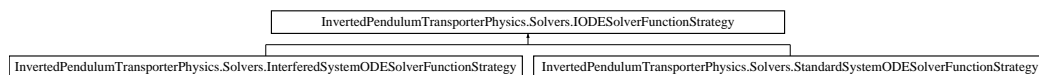
The documentation for this interface was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/PhysicsLibrary/Solvers/ODESolver.cs

## 5.12 InvertedPendulumTransporterPhysics.Solvers.IODESolverFunctionStrategy Interface Reference

Interface dedicated to ODE solving function strategies

Inheritance diagram for InvertedPendulumTransporterPhysics.Solvers.IODESolverFunctionStrategy:



### Public Member Functions

- void [ODESolverFunction](#) (double[] y, double x, double[] dy, object obj)  
*Function of solving the state-space equations in differential form*

### 5.12.1 Detailed Description

Interface dedicated to ODE solving function strategies

## 5.12.2 Member Function Documentation

### 5.12.2.1 void InvertedPendulumTransporterPhysics.Solvers.IODESolverFunctionStrategy.ODESolverFunction ( double[] y, double x, double[] dy, object obj )

Function of solving the state-space equations in differential form

Parameters

<i>y</i>	System state array
<i>x</i>	Time stamps array
<i>dy</i>	System state derivatives array
<i>obj</i>	Solver parameters

Implemented in [InvertedPendulumTransporterPhysics.Solvers.InterferedSystemODESolverFunctionStrategy](#), and [InvertedPendulumTransporterPhysics.Solvers.StandardSystemODESolverFunctionStrategy](#).

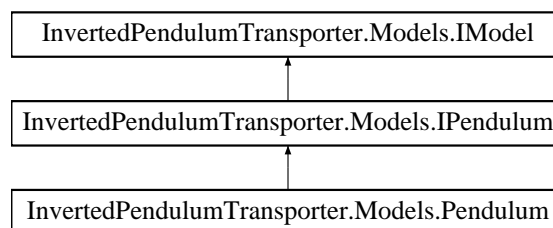
The documentation for this interface was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/PhysicsLibrary/Solvers/IODESolverFunctionStrategy.cs

## 5.13 InvertedPendulumTransporter.Models.IPendulum Interface Reference

Interface dedicated to pendulum model

Inheritance diagram for InvertedPendulumTransporter.Models.IPendulum:



### Properties

- Point3D [MassLinkPoint](#) [get]  
*Three dimensional point of pendulum mass link position*
- Point3D [CartLinkPoint](#) [get]  
*Three dimensional point of pendulum cart link position*
- double [RodLength](#) [get]  
*Pendulum rod length*

### Additional Inherited Members

#### 5.13.1 Detailed Description

Interface dedicated to pendulum model

#### 5.13.2 Property Documentation

#### 5.13.2.1 Point3D InvertedPendulumTransporter.Models.IPendulum.CartLinkPoint [get]

Three dimensional point of pendulum cart link position

#### 5.13.2.2 Point3D InvertedPendulumTransporter.Models.IPendulum.MassLinkPoint [get]

Three dimensional point of pendulum mass link position

#### 5.13.2.3 double InvertedPendulumTransporter.Models.IPendulum.RodLength [get]

Pendulum rod length

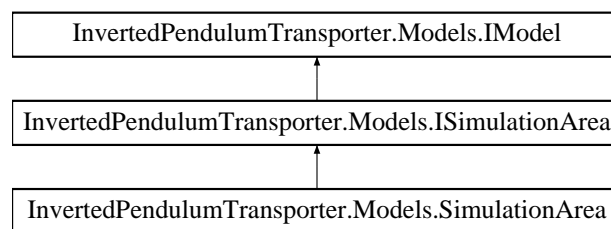
The documentation for this interface was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/Simulation/Models/IPendulum.cs

## 5.14 InvertedPendulumTransporter.Models.ISimulationArea Interface Reference

Interface dedicated to simulation area model

Inheritance diagram for InvertedPendulumTransporter.Models.ISimulationArea:



### Properties

- double [Size](#) [get]  
*Simulation area size*

### Additional Inherited Members

#### 5.14.1 Detailed Description

Interface dedicated to simulation area model

#### 5.14.2 Property Documentation

##### 5.14.2.1 double InvertedPendulumTransporter.Models.ISimulationArea.Size [get]

Simulation area size

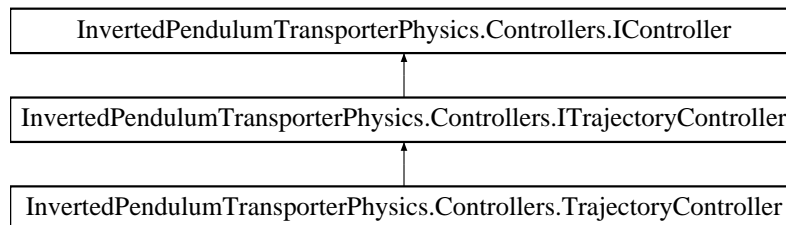
The documentation for this interface was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/Simulation/Models/ISimulationArea.cs

## 5.15 InvertedPendulumTransporterPhysics.Controllers.ITrajectoryController Interface Reference

Interface dedicated to trajectory controller

Inheritance diagram for InvertedPendulumTransporterPhysics.Controllers.ITrajectoryController:



### Public Member Functions

- void [Clear](#) ()  
*Clear trajectory*
- void [SetAccuracy](#) ([AccuracyType](#) accuracy)  
*Set trajectory tracking accuracy*
- [Point3DCollection](#) [LoadTrajectory](#) (string fileName=null)  
*Load trajectory from file*
- string [SaveTrajectory](#) (List< [Point3D](#) > trajectory)  
*Save trajectory to file*
- [Point3D](#) [GetTargetStartPosition](#) ()  
*Get trajectory beginning position*
- [Point3D](#) [GetTargetPosition](#) (double x, double y, out bool nextCheckPoint)  
*Get actual target control point*

### Properties

- bool [TrajectoryAchieved](#) [get]  
*Check if trajectory end was achieved*
- bool [TrajectoryEnabled](#) [get]  
*Check if trajectory mode is enabled*
- double [AverageDistance](#) [get]  
*Average distance between control points*

#### 5.15.1 Detailed Description

Interface dedicated to trajectory controller

#### 5.15.2 Member Function Documentation

##### 5.15.2.1 void InvertedPendulumTransporterPhysics.Controllers.ITrajectoryController.Clear ( )

Clear trajectory

Implemented in [InvertedPendulumTransporterPhysics.Controllers.TrajectoryController](#).



5.15.2.2 `Point3D` InvertedPendulumTransporterPhysics.Controllers.ITrajectoryController.GetTargetPosition ( `double x`, `double y`,  
out `bool nextCheckPoint` )

Get actual target control point

## Parameters

<i>x</i>	Cart position in X-coordinate
<i>y</i>	Cart position in Y-coordinate
<i>nextCheckPoint</i>	Info about target point change

## Returns

Target control point

Implemented in [InvertedPendulumTransporterPhysics.Controllers.TrajectoryController](#).

#### 5.15.2.3 Point3D InvertedPendulumTransporterPhysics.Controllers.ITrajectoryController.GetTargetStartPosition ( )

Get trajectory beginning position

## Returns

Implemented in [InvertedPendulumTransporterPhysics.Controllers.TrajectoryController](#).

#### 5.15.2.4 Point3DCollection InvertedPendulumTransporterPhysics.Controllers.ITrajectoryController.LoadTrajectory ( string *fileName* = null )

Load trajectory from file

## Parameters

<i>fileName</i>	File full path (if null additionally open file browser)
-----------------	---

## Returns

Control points collection with doubled points for trajectory visualization

Implemented in [InvertedPendulumTransporterPhysics.Controllers.TrajectoryController](#).

#### 5.15.2.5 string InvertedPendulumTransporterPhysics.Controllers.ITrajectoryController.SaveTrajectory ( List< Point3D > *trajectory* )

Save trajectory to file

## Parameters

<i>trajectory</i>	List of control points
-------------------	------------------------

## Returns

File full path

Implemented in [InvertedPendulumTransporterPhysics.Controllers.TrajectoryController](#).

#### 5.15.2.6 void InvertedPendulumTransporterPhysics.Controllers.ITrajectoryController.SetAccuracy ( AccuracyType *accuracy* )

Set trajectory tracking accuracy

## Parameters

<i>accuracy</i>	Accuracy type
-----------------	---------------

Implemented in [InvertedPendulumTransporterPhysics.Controllers.TrajectoryController](#).

### 5.15.3 Property Documentation

5.15.3.1 `double InvertedPendulumTransporterPhysics.Controllers.ITrajectoryController.AverageDistance` [\[get\]](#)

Average distance between control points

5.15.3.2 `bool InvertedPendulumTransporterPhysics.Controllers.ITrajectoryController.TrajectoryAchieved` [\[get\]](#)

Check if trajectory end was achieved

5.15.3.3 `bool InvertedPendulumTransporterPhysics.Controllers.ITrajectoryController.TrajectoryEnabled` [\[get\]](#)

Check if trajectory mode is enabled

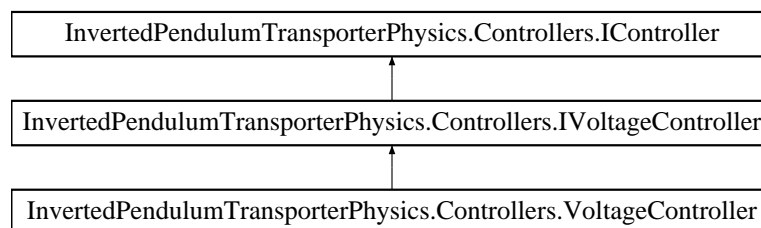
The documentation for this interface was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/PhysicsLibrary/Controllers/ITrajectoryController.cs

## 5.16 InvertedPendulumTransporterPhysics.Controllers.IVoltageController Interface Reference

Interface dedicated to voltage controller

Inheritance diagram for InvertedPendulumTransporterPhysics.Controllers.IVoltageController:



### Public Member Functions

- `double GetVoltage ()`  
*Get result voltage from regulation process*
- `void SetTime (double time)`  
*Set current simulation time (for sinusoidal methods)*
- `void SetControlError (double angleError, double positionError)`  
*Set control error*
- `void Reset (double timeDelta)`  
*Reset controller with given time delta*
- `void SetUserAngle (double userAngle)`  
*Setup angle generated by user (for none control)*

## Properties

- [ControlType ControlType](#) [get, set]  
*Type of voltage controller*

### 5.16.1 Detailed Description

Interface dedicated to voltage controller

### 5.16.2 Member Function Documentation

#### 5.16.2.1 `double InvertedPendulumTransporterPhysics.Controllers.IVoltageController.GetVoltage ( )`

Get result voltage from regulation process

Returns

Motor Voltage

Implemented in [InvertedPendulumTransporterPhysics.Controllers.VoltageController](#).

#### 5.16.2.2 `void InvertedPendulumTransporterPhysics.Controllers.IVoltageController.Reset ( double timeDelta )`

Reset controller with given time delta

Parameters

<i>timeDelta</i>	Actual time delta
------------------	-------------------

Implemented in [InvertedPendulumTransporterPhysics.Controllers.VoltageController](#).

#### 5.16.2.3 `void InvertedPendulumTransporterPhysics.Controllers.IVoltageController.SetControlError ( double angleError, double positionError )`

Set control error

Parameters

<i>angleError</i>	Difference between desired angle and actual one
<i>positionError</i>	Difference between desired position and actual one

Implemented in [InvertedPendulumTransporterPhysics.Controllers.VoltageController](#).

#### 5.16.2.4 `void InvertedPendulumTransporterPhysics.Controllers.IVoltageController.SetTime ( double time )`

Set current simulation time (for sinusoidal methods)

Parameters

<i>time</i>	Current simulation time
-------------	-------------------------

Implemented in [InvertedPendulumTransporterPhysics.Controllers.VoltageController](#).

#### 5.16.2.5 `void InvertedPendulumTransporterPhysics.Controllers.IVoltageController.SetUserAngle ( double userAngle )`

Setup angle generated by user (for none control)

## Parameters

<i>userAngle</i>	User angle value
------------------	------------------

Implemented in [InvertedPendulumTransporterPhysics.Controllers.VoltageController](#).

### 5.16.3 Property Documentation

5.16.3.1 **ControlType** [InvertedPendulumTransporterPhysics.Controllers.IVoltageController.ControlType](#) [get], [set]

Type of voltage controller

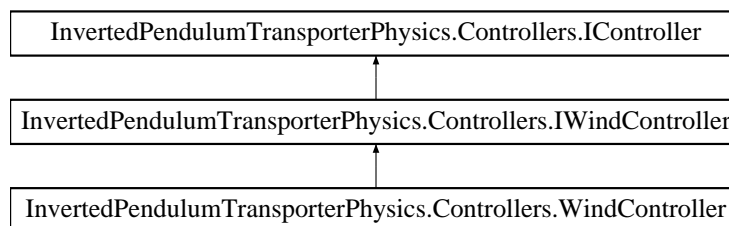
The documentation for this interface was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/PhysicsLibrary/Controllers/IVoltageController.cs

## 5.17 InvertedPendulumTransporterPhysics.Controllers.IWindController Interface Reference

Interface dedicated to wind controller

Inheritance diagram for [InvertedPendulumTransporterPhysics.Controllers.IWindController](#):



### Public Member Functions

- [Vector3D UpdateWindForce](#) ()  
*Update wind force*
- [double GetXCoordWindPower](#) ()  
*Get final wind power in X-coordinate*
- [double GetYCoordWindPower](#) ()  
*Get final wind power in X-coordinate*
- [double GetZCoordWindPower](#) ()  
*Get final wind power in X-coordinate*

### Properties

- [WindType WindType](#) [get, set]  
*Wind type*
- [double WindPower](#) [get, set]  
*Wind power*
- [double WindChangeSpeed](#) [get, set]  
*Wind change speed*
- [double MaxWindPower](#) [get]  
*Max wind power*

- double [MinWindPower](#) [get]  
*Min wind power*
- double [DefaultWindPower](#) [get]  
*Default wind power*

### 5.17.1 Detailed Description

Interface dedicated to wind controller

### 5.17.2 Member Function Documentation

#### 5.17.2.1 double [InvertedPendulumTransporterPhysics.Controllers.IWindController.GetXCoordWindPower](#) ( )

Get final wind power in X-coordinate

Returns

Wind power

Implemented in [InvertedPendulumTransporterPhysics.Controllers.WindController](#).

#### 5.17.2.2 double [InvertedPendulumTransporterPhysics.Controllers.IWindController.GetYCoordWindPower](#) ( )

Get final wind power in X-coordinate

Returns

Wind power

Implemented in [InvertedPendulumTransporterPhysics.Controllers.WindController](#).

#### 5.17.2.3 double [InvertedPendulumTransporterPhysics.Controllers.IWindController.GetZCoordWindPower](#) ( )

Get final wind power in X-coordinate

Returns

Wind power

Implemented in [InvertedPendulumTransporterPhysics.Controllers.WindController](#).

#### 5.17.2.4 [Vector3D](#) [InvertedPendulumTransporterPhysics.Controllers.IWindController.UpdateWindForce](#) ( )

Update wind force

Returns

Wind force direction

Implemented in [InvertedPendulumTransporterPhysics.Controllers.WindController](#).

### 5.17.3 Property Documentation

#### 5.17.3.1 double [InvertedPendulumTransporterPhysics.Controllers.IWindController.DefaultWindPower](#) [get]

Default wind power

5.17.3.2 `double InvertedPendulumTransporterPhysics.Controllers.IWindController.MaxWindPower` `[get]`

Max wind power

5.17.3.3 `double InvertedPendulumTransporterPhysics.Controllers.IWindController.MinWindPower` `[get]`

Min wind power

5.17.3.4 `double InvertedPendulumTransporterPhysics.Controllers.IWindController.WindChangeSpeed` `[get], [set]`

Wind change speed

5.17.3.5 `double InvertedPendulumTransporterPhysics.Controllers.IWindController.WindPower` `[get], [set]`

Wind power

5.17.3.6 `WindType InvertedPendulumTransporterPhysics.Controllers.IWindController.WindType` `[get], [set]`

Wind type

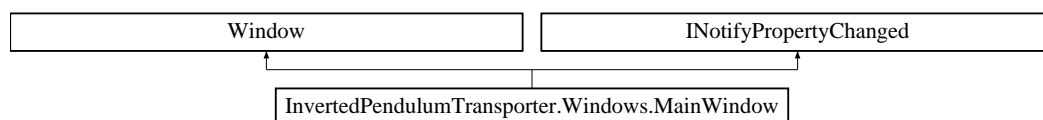
The documentation for this interface was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/PhysicsLibrary/Controllers/IWindController.cs

## 5.18 InvertedPendulumTransporter.Windows.MainWindow Class Reference

Interaction logic for MainWindow.xaml [application management]

Inheritance diagram for InvertedPendulumTransporter.Windows.MainWindow:



### Public Member Functions

- [MainWindow\(\)](#)  
*Class constructor*

### Properties

- `double` [MaxAngle](#) `[get]`  
*Max pendulum angle*
- `double` [MinAngle](#) `[get]`  
*Min pendulum angle*
- `double` [MaxWindPower](#) `[get]`  
*Max wind power*
- `double` [MinWindPower](#) `[get]`  
*Min wind power*

- double [CartPositionX](#) [get]  
*Actual cart position in X-coordinate*
- double [CartPositionY](#) [get]  
*Actual cart position in Y-coordinate*
- double [CartVelocityX](#) [get]  
*Actual cart velocity in X-coordinate*
- double [CartVelocityY](#) [get]  
*Actual cart velocity in Y-coordinate*
- double [PendulumAngleX](#) [get]  
*Actual pendulum angle in X-coordinate*
- double [PendulumAngleY](#) [get]  
*Actual pendulum angle in Y-coordinate*
- double [PendulumAngularVelocityX](#) [get]  
*Actual pendulum angular velocity in X-coordinate*
- double [PendulumAngularVelocityY](#) [get]  
*Actual pendulum angular velocity in Y-coordinate*
- double [TimeDelta](#) [get, set]  
*Actual time distance between two frames*
- double [XCoordAngle](#) [get, set]  
*Initial pendulum angle in X-coordinate*
- double [YCoordAngle](#) [get, set]  
*Initial pendulum angle in Y-coordinate*
- double [RodLength](#) [get, set]  
*Actual pendulum rod length*
- double [PendulumMass](#) [get, set]  
*Actual pendulum mass*
- double [CartMass](#) [get, set]  
*Actual cart mass*
- double [WindPower](#) [get, set]  
*Actual wind power*
- double [WindChangeSpeed](#) [get, set]  
*Actual wind change speed*

## Events

- PropertyChangedEventHandler [PropertyChanged](#)  
*Event handler for raising property change*

### 5.18.1 Detailed Description

Interaction logic for MainWindow.xaml [application management]

Interaction logic for MainWindow.xaml [Top Menu management]

Interaction logic for MainWindow.xaml [Control Panel management]

### 5.18.2 Constructor & Destructor Documentation

#### 5.18.2.1 InvertedPendulumTransporter.Windows.MainWindow.MainWindow ( ) [inline]

Class constructor



### 5.18.3 Property Documentation

5.18.3.1 `double InvertedPendulumTransporter.Windows.MainWindow.CartMass` [get], [set]

Actual cart mass

5.18.3.2 `double InvertedPendulumTransporter.Windows.MainWindow.CartPositionX` [get]

Actual cart position in X-coordinate

5.18.3.3 `double InvertedPendulumTransporter.Windows.MainWindow.CartPositionY` [get]

Actual cart position in Y-coordinate

5.18.3.4 `double InvertedPendulumTransporter.Windows.MainWindow.CartVelocityX` [get]

Actual cart velocity in X-coordinate

5.18.3.5 `double InvertedPendulumTransporter.Windows.MainWindow.CartVelocityY` [get]

Actual cart velocity in Y-coordinate

5.18.3.6 `double InvertedPendulumTransporter.Windows.MainWindow.MaxAngle` [get]

Max pendulum angle

5.18.3.7 `double InvertedPendulumTransporter.Windows.MainWindow.MaxWindPower` [get]

Max wind power

5.18.3.8 `double InvertedPendulumTransporter.Windows.MainWindow.MinAngle` [get]

Min pendulum angle

5.18.3.9 `double InvertedPendulumTransporter.Windows.MainWindow.MinWindPower` [get]

Min wind power

5.18.3.10 `double InvertedPendulumTransporter.Windows.MainWindow.PendulumAngleX` [get]

Actual pendulum angle in X-coordinate

5.18.3.11 `double InvertedPendulumTransporter.Windows.MainWindow.PendulumAngleY` [get]

Actual pendulum angle in Y-coordinate

5.18.3.12 `double InvertedPendulumTransporter.Windows.MainWindow.PendulumAngularVelocityX` [get]

Actual pendulum angular velocity in X-coordinate

5.18.3.13 `double InvertedPendulumTransporter.Windows.MainWindow.PendulumAngularVelocityY` [get]

Actual pendulum angular velocity in Y-coordinate

5.18.3.14 `double InvertedPendulumTransporter.Windows.MainWindow.PendulumMass` [get], [set]

Actual pendulum mass

5.18.3.15 `double InvertedPendulumTransporter.Windows.MainWindow.RodLength` [get], [set]

Actual pendulum rod length

5.18.3.16 `double InvertedPendulumTransporter.Windows.MainWindow.TimeDelta` [get], [set]

Actual time distance between two frames

5.18.3.17 `double InvertedPendulumTransporter.Windows.MainWindow.WindChangeSpeed` [get], [set]

Actual wind change speed

5.18.3.18 `double InvertedPendulumTransporter.Windows.MainWindow.WindPower` [get], [set]

Actual wind power

5.18.3.19 `double InvertedPendulumTransporter.Windows.MainWindow.XCoordAngle` [get], [set]

Initial pendulum angle in X-coordinate

5.18.3.20 `double InvertedPendulumTransporter.Windows.MainWindow.YCoordAngle` [get], [set]

Initial pendulum angle in Y-coordinate

## 5.18.4 Event Documentation

5.18.4.1 `PropertyChangedEventHandler InvertedPendulumTransporter.Windows.MainWindow.PropertyChanged`

Event handler for raising property change

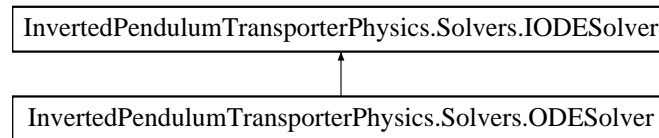
The documentation for this class was generated from the following files:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/Simulation/Windows/MainWindow.xaml.cs
- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/Simulation/Windows/MainWindowControlPanel.cs
- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/Simulation/Windows/MainWindowMenu.cs

## 5.19 InvertedPendulumTransporterPhysics.Solvers.ODESolver Class Reference

ODE solver class

Inheritance diagram for InvertedPendulumTransporterPhysics.Solvers.ODESolver:



## Public Member Functions

- [ODESolver](#) ([SolverParameters](#) parameters)  
*Class constructor*
- void [SetupStrategy](#) ([IODESolverFunctionStrategy](#) strategy)  
*Setup solving strategy*
- void [UpdateSystemParameters](#) ([SolverParameters](#) parameters)  
*Update solver parameters*
- [OneDimensionalSystemState](#) [SolveODESystem](#) (double[] x, double[] y)  
*Solve ordinary differential equation (ODE)*

### 5.19.1 Detailed Description

ODE solver class

### 5.19.2 Constructor & Destructor Documentation

5.19.2.1 [InvertedPendulumTransporterPhysics.Solvers.ODESolver.ODESolver](#) ( [SolverParameters](#) parameters )  
[inline]

Class constructor

Parameters

<i>parameters</i>	Solver parameters
-------------------	-------------------

### 5.19.3 Member Function Documentation

5.19.3.1 void [InvertedPendulumTransporterPhysics.Solvers.ODESolver.SetupStrategy](#) ( [IODESolverFunctionStrategy](#) strategy ) [inline]

Setup solving strategy

Parameters

<i>strategy</i>	Strategy of solving equations
-----------------	-------------------------------

Implements [InvertedPendulumTransporterPhysics.Solvers.IODESolver](#).

5.19.3.2 [OneDimensionalSystemState](#) [InvertedPendulumTransporterPhysics.Solvers.ODESolver.SolveODESystem](#) ( double[] x, double[] y ) [inline]

Solve ordinary differential equation (ODE)

## Parameters

<i>x</i>	System state array
<i>y</i>	Time stamps array

## Returns

New system state

Implements [InvertedPendulumTransporterPhysics.Solvers.IODESolver](#).

5.19.3.3 `void InvertedPendulumTransporterPhysics.Solvers.ODESolver.UpdateSystemParameters ( SolverParameters parameters ) [inline]`

Update solver parameters

## Parameters

<i>parameters</i>	New solver parameters
-------------------	-----------------------

Implements [InvertedPendulumTransporterPhysics.Solvers.IODESolver](#).

The documentation for this class was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/PhysicsLibrary/Solvers/ODESolver.cs

## 5.20 InvertedPendulumTransporterPhysics.Common.OneDimensionalSystemState Class Reference

Container for subsystem state

## Public Member Functions

- [OneDimensionalSystemState](#) ()  
*Class constructor*
- [OneDimensionalSystemState](#) (double position, double angle, double velocity, double angularVelocity)  
*Class constructor with initialization*
- void [InitializeDefault](#) ()  
*Initialization with default parameters (except angle)*
- double[] [ToStateArray](#) ()  
*Convert system state to array*

## Properties

- double [Position](#) [get, set]  
*Cart position on axis*
- double [Velocity](#) [get, set]  
*Cart velocity*
- double [Angle](#) [get, set]  
*Pendulum angle*
- double [AngularVelocity](#) [get, set]  
*Pendulum angular velocity*

### 5.20.1 Detailed Description

Container for subsystem state

### 5.20.2 Constructor & Destructor Documentation

5.20.2.1 `InvertedPendulumTransporterPhysics.Common.OneDimensionalSystemState.OneDimensionalSystemState ( )` `[inline]`

Class constructor

5.20.2.2 `InvertedPendulumTransporterPhysics.Common.OneDimensionalSystemState.OneDimensionalSystemState ( double position, double angle, double velocity, double angularVelocity )` `[inline]`

Class constructor with initialization

Parameters

<i>position</i>	Initial cart position
<i>angle</i>	Initial pendulum angle
<i>velocity</i>	Initial cart velocity
<i>angularVelocity</i>	Initial pendulum angular velocity

### 5.20.3 Member Function Documentation

5.20.3.1 `void InvertedPendulumTransporterPhysics.Common.OneDimensionalSystemState.InitializeDefault ( )` `[inline]`

Initialization with default parameters (except angle)

5.20.3.2 `double [] InvertedPendulumTransporterPhysics.Common.OneDimensionalSystemState.ToStateArray ( )` `[inline]`

Convert system state to array

Returns

Array of system parameters

### 5.20.4 Property Documentation

5.20.4.1 `double InvertedPendulumTransporterPhysics.Common.OneDimensionalSystemState.Angle` `[get]`, `[set]`

Pendulum angle

5.20.4.2 `double InvertedPendulumTransporterPhysics.Common.OneDimensionalSystemState.AngularVelocity` `[get]`, `[set]`

Pendulum angular velocity

5.20.4.3 `double InvertedPendulumTransporterPhysics.Common.OneDimensionalSystemState.Position` `[get]`, `[set]`

Cart position on axis

5.20.4.4 `double InvertedPendulumTransporterPhysics.Common.OneDimensionalSystemState.Velocity` `[get]`, `[set]`

Cart velocity

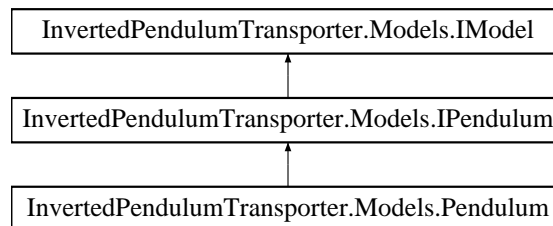
The documentation for this class was generated from the following file:

- `C:/Users/Arkady/Downloads/Murzyński Backup/BIB/PhysicsLibrary/Common/OneDimensionalSystemState.cs`

## 5.21 InvertedPendulumTransporter.Models.Pendulum Class Reference

[Pendulum](#) model class

Inheritance diagram for `InvertedPendulumTransporter.Models.Pendulum`:



### Public Member Functions

- [Pendulum](#) ()  
*Class constructor*
- void [Initialize](#) ()  
*Initialization method*
- void [UpdateState](#) ([SystemState](#) systemState)  
*Uptate model visual*
- void [SetupHighLevelGraphics](#) ()  
*Seup high level graphics for model*
- void [SetupLowLevelGraphics](#) ()  
*Setup low level graphics for model*

### Properties

- `ModelVisual3D` **Model** `[get]`
- `Point3D` **MassLinkPoint** `[get]`
- `Point3D` **CartLinkPoint** `[get]`
- `double` **RodLength** `[get]`

#### 5.21.1 Detailed Description

[Pendulum](#) model class

#### 5.21.2 Constructor & Destructor Documentation

5.21.2.1 `InvertedPendulumTransporter.Models.Pendulum.Pendulum ( )` `[inline]`

Class constructor

### 5.21.3 Member Function Documentation

#### 5.21.3.1 void InvertedPendulumTransporter.Models.Pendulum.Initialize ( ) [inline]

Initialization method

Implements [InvertedPendulumTransporter.Models.IModel](#).

#### 5.21.3.2 void InvertedPendulumTransporter.Models.Pendulum.SetupHighLevelGraphics ( ) [inline]

Seup high level graphics for model

Implements [InvertedPendulumTransporter.Models.IModel](#).

#### 5.21.3.3 void InvertedPendulumTransporter.Models.Pendulum.SetupLowLevelGraphics ( ) [inline]

Setup low level graphics for model

Implements [InvertedPendulumTransporter.Models.IModel](#).

#### 5.21.3.4 void InvertedPendulumTransporter.Models.Pendulum.UpdateState ( SystemState *systemState* ) [inline]

Uptate model visual

Parameters

<i>systemState</i>	Actual system state
--------------------	---------------------

Implements [InvertedPendulumTransporter.Models.IModel](#).

The documentation for this class was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/Simulation/Models/Pendulum.cs

## 5.22 InvertedPendulumTransporterPhysics.Controllers.PIDCorrector Class Reference

Helping class for voltage controlling dedicated to PID corrector

### Public Member Functions

- [PIDCorrector](#) ()  
*Class constructor*
- void [SetAngleError](#) (double error)  
*Setup pendulum angle error*
- void [SetPositionError](#) (double error)  
*Setup cart position error*
- void [Reset](#) ()  
*Reset object to default state*
- void [Reset](#) (double timeDelta)  
*Reset object to default state with setting new time delta*
- double [CalculateAnglePIDCorrection](#) ()  
*Calculate PID correction for pendulum angle*
- double [CalculatePositionPIDCorrection](#) (double angleError)  
*Calculate PID correction for cart position*
- double [CalculateParallelPositionAnglePIDCorrection](#) (bool integral)  
*Calculate PID correction parallely for cart position and pendulum angle*

### 5.22.1 Detailed Description

Helping class for voltage controlling dedicated to PID corrector

### 5.22.2 Constructor & Destructor Documentation

#### 5.22.2.1 InvertedPendulumTransporterPhysics.Controllers.PIDCorrector.PIDCorrector ( ) [inline]

Class constructor

### 5.22.3 Member Function Documentation

#### 5.22.3.1 double InvertedPendulumTransporterPhysics.Controllers.PIDCorrector.CalculateAnglePIDCorrection ( ) [inline]

Calculate PID correction for pendulum angle

Returns

Motor voltage

#### 5.22.3.2 double InvertedPendulumTransporterPhysics.Controllers.PIDCorrector.CalculateParallelPositionAnglePIDCorrection ( bool *integral* ) [inline]

Calculate PID correction parallelly for cart position and pendulum angle

Parameters

<i>integral</i>	Use integral part in pendulum angle correction
-----------------	--

Returns

#### 5.22.3.3 double InvertedPendulumTransporterPhysics.Controllers.PIDCorrector.CalculatePositionPIDCorrection ( double *angleError* ) [inline]

Calculate PID correction for cart position

Parameters

<i>angleError</i>	Pendulum angle error
-------------------	----------------------

Returns

Motor voltage

#### 5.22.3.4 void InvertedPendulumTransporterPhysics.Controllers.PIDCorrector.Reset ( ) [inline]

Reset object to default state

#### 5.22.3.5 void InvertedPendulumTransporterPhysics.Controllers.PIDCorrector.Reset ( double *timeDelta* ) [inline]

Reset object to default state with setting new time delta



## Parameters

<i>timeDelta</i>	Time delta
------------------	------------

5.22.3.6 void InvertedPendulumTransporterPhysics.Controllers.PIDCorrector.SetAngleError ( double *error* ) [inline]

Setup pendulum angle error

## Parameters

<i>error</i>	Angle error
--------------	-------------

5.22.3.7 void InvertedPendulumTransporterPhysics.Controllers.PIDCorrector.SetPositionError ( double *error* ) [inline]

Setup cart position error

## Parameters

<i>error</i>	Position error
--------------	----------------

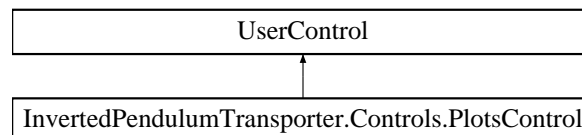
The documentation for this class was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/PhysicsLibrary/Controllers/PIDCorrector.cs

## 5.23 InvertedPendulumTransporter.Controls.PlotsControl Class Reference

Interaction logic for PlotsControl.xaml

Inheritance diagram for InvertedPendulumTransporter.Controls.PlotsControl:



### Public Member Functions

- [PlotsControl](#) ()  
*Class constructor*
- void [UpdateVoltagePlots](#) (double time, double xCoordVoltage, double yCoordVoltage)  
*Update voltage plot data*
- void [UpdateAngleErrorPlots](#) (double time, double xCoordError, double yCoordError)  
*Update angle error plot data*
- void [UpdatePositionErrorPlots](#) (double time, double xCoordError, double yCoordError)  
*Update position error plot data*
- void [ResetPlots](#) ()  
*Reset plots state to default*
- void [PassParameters](#) (double timeDelta, double xCoordAngle, double yCoordAngle, double rodLength, double cartMass, double pendulumMass, double windPower)  
*Pass actual parameters for file save info*

## Properties

- `ObservableCollection< DataPoint > AngleErrorPointsX` [get]  
*Collection for pendulum angle errors in X-coordinate*
- `ObservableCollection< DataPoint > PositionErrorPointsX` [get]  
*Collection for cart position errors in X-coordinate*
- `ObservableCollection< DataPoint > VoltagePointsX` [get]  
*Collection for motor voltage values in X-coordinate*
- `ObservableCollection< DataPoint > AngleErrorPointsY` [get]  
*Collection for cart pendulum angle errors in Y-coordinate*
- `ObservableCollection< DataPoint > PositionErrorPointsY` [get]  
*Collection for cart position errors in Y-coordinate*
- `ObservableCollection< DataPoint > VoltagePointsY` [get]  
*Collection for motor voltage values in Y-coordinate*

### 5.23.1 Detailed Description

Interaction logic for PlotsControl.xaml

### 5.23.2 Constructor & Destructor Documentation

#### 5.23.2.1 `InvertedPendulumTransporter.Controls.PlotsControl.PlotsControl ( )` [inline]

Class constructor

### 5.23.3 Member Function Documentation

#### 5.23.3.1 `void InvertedPendulumTransporter.Controls.PlotsControl.PassParameters ( double timeDelta, double xCoordAngle, double yCoordAngle, double rodLength, double cartMass, double pendulumMass, double windPower )` [inline]

Pass actual parameters for file save info

Parameters

<i>timeDelta</i>	Time delta
<i>xCoordAngle</i>	Initial angle in X-coordinate
<i>yCoordAngle</i>	Initial angle in Y-coordinate
<i>rodLength</i>	Rod length
<i>cartMass</i>	Cart mass
<i>pendulumMass</i>	Pendulum mass
<i>windPower</i>	Wind power

#### 5.23.3.2 `void InvertedPendulumTransporter.Controls.PlotsControl.ResetPlots ( )` [inline]

Reset plots state to default

#### 5.23.3.3 `void InvertedPendulumTransporter.Controls.PlotsControl.UpdateAngleErrorPlots ( double time, double xCoordError, double yCoordError )` [inline]

Update angle error plot data

## Parameters

<i>time</i>	Animation time
<i>xCoordError</i>	Angle error in X-coordinate
<i>yCoordError</i>	Angle error in Y-coordinate

5.23.3.4 `void InvertedPendulumTransporter.Controls.PlotsControl.UpdatePositionErrorPlots ( double time, double xCoordError, double yCoordError ) [inline]`

Update position error plot data

## Parameters

<i>time</i>	Animation time
<i>xCoordError</i>	Position error in X-coordinate
<i>yCoordError</i>	Position error in Y-coordinate

5.23.3.5 `void InvertedPendulumTransporter.Controls.PlotsControl.UpdateVoltagePlots ( double time, double xCoordVoltage, double yCoordVoltage ) [inline]`

Update voltage plot data

## Parameters

<i>time</i>	Animation time
<i>xCoordVoltage</i>	Voltage value in X-coordinate
<i>yCoordVoltage</i>	Voltage value in Y-coordinate

## 5.23.4 Property Documentation

5.23.4.1 `ObservableCollection<DataPoint> InvertedPendulumTransporter.Controls.PlotsControl.AngleErrorPointsX [get]`

Collection for pendulum angle errors in X-coordinate

5.23.4.2 `ObservableCollection<DataPoint> InvertedPendulumTransporter.Controls.PlotsControl.AngleErrorPointsY [get]`

Collection for cart pendulum angle errors in Y-coordinate

5.23.4.3 `ObservableCollection<DataPoint> InvertedPendulumTransporter.Controls.PlotsControl.PositionErrorPointsX [get]`

Collection for cart position errors in X-coordinate

5.23.4.4 `ObservableCollection<DataPoint> InvertedPendulumTransporter.Controls.PlotsControl.PositionErrorPointsY [get]`

Collection for cart position errors in Y-coordinate

5.23.4.5 `ObservableCollection<DataPoint> InvertedPendulumTransporter.Controls.PlotsControl.VoltagePointsX [get]`

Collection for motor voltage values in X-coordinate

#### 5.23.4.6 ObservableCollection<DataPoint> InvertedPendulumTransporter.Controls.PlotsControl.VoltagePointsY [get]

Collection for motor voltage values in Y-coordinate

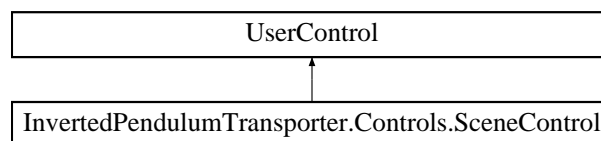
The documentation for this class was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/Simulation/Controls/PlotsControl.xaml.cs

## 5.24 InvertedPendulumTransporter.Controls.SceneControl Class Reference

Interaction logic for Scene.xaml

Inheritance diagram for InvertedPendulumTransporter.Controls.SceneControl:



### Public Member Functions

- [SceneControl](#) ()  
*Class constructor*
- void [SetupHighLevelGraphics](#) ()  
*Setup high level graphics*
- void [SetupLowLevelGraphics](#) ()  
*Setup low level graphics*
- void [UpdateState](#) (SystemState systemState)  
*Update scene models*
- void [UpdateCamera](#) (SystemState systemState)  
*Update camera object*
- void [UpdateFrame](#) (SystemState systemState)  
*Update rendering frame*
- void [ResetSimulation](#) (SystemState systemState)  
*Reset scene to start configuration*
- void [UpdateWindDirection](#) (Vector3D windDirection, double windPower)  
*Update wind direction visual*
- void [ClearTrajectory](#) ()  
*Clear trajectories*
- void [UpdateTrajectory](#) (Point3DCollection trajectory)  
*Update target trajectory*
- void [ShowTargetTrajectory](#) (bool isChecked)  
*Change target trajectory visibility*
- void [ShowCartTrajectory](#) (bool isChecked)  
*Change cart trajectory visibility*
- void [ShowPendulumTrajectory](#) (bool isChecked)  
*Change pendulum trajectory visibility*

## Public Attributes

- [IPendulum pendulum](#)  
*Reference to pendulum model*
- [ICart cart](#)  
*Reference to cart model*
- [ISimulationArea simulationArea](#)  
*Reference to simulation area model*

### 5.24.1 Detailed Description

Interaction logic for Scene.xaml

### 5.24.2 Constructor & Destructor Documentation

5.24.2.1 `InvertedPendulumTransporter.Controls.SceneControl.SceneControl ( )` [\[inline\]](#)

Class constructor

### 5.24.3 Member Function Documentation

5.24.3.1 `void InvertedPendulumTransporter.Controls.SceneControl.ClearTrajectory ( )` [\[inline\]](#)

Clear trajectories

5.24.3.2 `void InvertedPendulumTransporter.Controls.SceneControl.ResetSimulation ( SystemState systemState )`  
[\[inline\]](#)

Reset scene to start configuration

Parameters

<i>systemState</i>	Actual system state
--------------------	---------------------

5.24.3.3 `void InvertedPendulumTransporter.Controls.SceneControl.SetupHighLevelGraphics ( )` [\[inline\]](#)

Setup high level graphics

5.24.3.4 `void InvertedPendulumTransporter.Controls.SceneControl.SetupLowLevelGraphics ( )` [\[inline\]](#)

Setup low level graphics

5.24.3.5 `void InvertedPendulumTransporter.Controls.SceneControl.ShowCartTrajectory ( bool isChecked )` [\[inline\]](#)

Change cart trajectory visibility

Parameters

<i>isChecked</i>	Is trajectory visible
------------------	-----------------------

5.24.3.6 **void** InvertedPendulumTransporter.Controls.SceneControl.ShowPendulumTrajectory ( **bool** *isChecked* )  
[inline]

Change pendulum trajectory visibility

Parameters

<i>isChecked</i>	Is trajectory visible
------------------	-----------------------

5.24.3.7 **void** InvertedPendulumTransporter.Controls.SceneControl.ShowTargetTrajectory ( **bool** *isChecked* ) [inline]

Change target trajectory visibility

Parameters

<i>isChecked</i>	Is trajectory visible
------------------	-----------------------

5.24.3.8 **void** InvertedPendulumTransporter.Controls.SceneControl.UpdateCamera ( **SystemState** *systemState* )  
[inline]

Update camera object

Parameters

<i>systemState</i>	Actual system state
--------------------	---------------------

5.24.3.9 **void** InvertedPendulumTransporter.Controls.SceneControl.UpdateFrame ( **SystemState** *systemState* )  
[inline]

Update rendering frame

Parameters

<i>systemState</i>	Actual system staate
--------------------	----------------------

5.24.3.10 **void** InvertedPendulumTransporter.Controls.SceneControl.UpdateState ( **SystemState** *systemState* )  
[inline]

Update scene models

Parameters

<i>systemState</i>	
--------------------	--

5.24.3.11 **void** InvertedPendulumTransporter.Controls.SceneControl.UpdateTrajectory ( **Point3DCollection** *trajectory* )  
[inline]

Update target trajectory

## Parameters

<i>trajectory</i>	Trajectory
-------------------	------------

5.24.3.12 void InvertedPendulumTransporter.Controls.SceneControl.UpdateWindDirection ( Vector3D *windDirection*, double *windPower* ) [inline]

Update wind direction visual

## Parameters

<i>windDirection</i>	Wind direction vector
<i>windPower</i>	Wind power value

## 5.24.4 Member Data Documentation

5.24.4.1 ICart InvertedPendulumTransporter.Controls.SceneControl.cart

Reference to cart model

5.24.4.2 IPendulum InvertedPendulumTransporter.Controls.SceneControl.pendulum

Reference to pendulum model

5.24.4.3 ISimulationArea InvertedPendulumTransporter.Controls.SceneControl.simulationArea

Reference to simulation area model

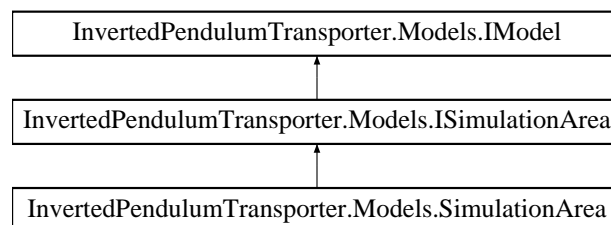
The documentation for this class was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/Simulation/Controls/SceneControl.xaml.cs

## 5.25 InvertedPendulumTransporter.Models.SimulationArea Class Reference

Simulation area model class

Inheritance diagram for InvertedPendulumTransporter.Models.SimulationArea:



## Public Member Functions

- [SimulationArea](#) ()  
*Class constructor*
- void [Initialize](#) ()  
*Initialization method*

- void [UpdateState](#) ([SystemState](#) systemState)  
*Uptate model visual*
- void [SetupHighLevelGraphics](#) ()  
*Seup high level graphics for model*
- void [SetupLowLevelGraphics](#) ()  
*Setup low level graphics for model*

## Properties

- double **Size** [get]
- [ModelVisual3D](#) **Model** [get]

### 5.25.1 Detailed Description

Simulation area model class

### 5.25.2 Constructor & Destructor Documentation

5.25.2.1 [InvertedPendulumTransporter.Models.SimulationArea.SimulationArea](#) ( ) [inline]

Class constructor

### 5.25.3 Member Function Documentation

5.25.3.1 void [InvertedPendulumTransporter.Models.SimulationArea.Initialize](#) ( ) [inline]

Initialization method

Implements [InvertedPendulumTransporter.Models.IModel](#).

5.25.3.2 void [InvertedPendulumTransporter.Models.SimulationArea.SetupHighLevelGraphics](#) ( ) [inline]

Seup high level graphics for model

Implements [InvertedPendulumTransporter.Models.IModel](#).

5.25.3.3 void [InvertedPendulumTransporter.Models.SimulationArea.SetupLowLevelGraphics](#) ( ) [inline]

Setup low level graphics for model

Implements [InvertedPendulumTransporter.Models.IModel](#).

5.25.3.4 void [InvertedPendulumTransporter.Models.SimulationArea.UpdateState](#) ( [SystemState](#) systemState )  
[inline]

Uptate model visual

Parameters

---



<code>systemState</code>	Actual system state
--------------------------	---------------------

Implements [InvertedPendulumTransporter.Models.IModel](#).

The documentation for this class was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/Simulation/Models/SimulationArea.cs

## 5.26 InvertedPendulumTransporterPhysics.Common.SolverParameters Class Reference

Container for solver parameters

### Public Member Functions

- [SolverParameters](#) ()  
*Class constructor*

### Public Attributes

- const double [G](#) = 9.83  
*Gravity acceleration*

### Properties

- double [PendulumMass](#) [get, set]  
*Pendulum mass*
- double [CartMass](#) [get, set]  
*Cart mass*
- double [PendulumLength](#) [get, set]  
*Pendulum Length*
- double [Voltage](#) [get, set]  
*Motor voltage*
- double [Gamma1](#) [get, set]  
*Cart friction factor*
- double [Gamma2](#) [get, set]  
*Voltage conversion factor*
- double [HorizontalWindForce](#) [get, set]  
*Horizontal wind force*
- double [VerticalWindForce](#) [get, set]  
*Vertical wind force*

#### 5.26.1 Detailed Description

Container for solver parameters

#### 5.26.2 Constructor & Destructor Documentation

##### 5.26.2.1 InvertedPendulumTransporterPhysics.Common.SolverParameters.SolverParameters ( ) [inline]

Class constructor

### 5.26.3 Member Data Documentation

5.26.3.1 `const double InvertedPendulumTransporterPhysics.Common.SolverParameters.G = 9.83`

Gravity acceleration

### 5.26.4 Property Documentation

5.26.4.1 `double InvertedPendulumTransporterPhysics.Common.SolverParameters.CartMass` [get], [set]

Cart mass

5.26.4.2 `double InvertedPendulumTransporterPhysics.Common.SolverParameters.Gamma1` [get], [set]

Cart friction factor

5.26.4.3 `double InvertedPendulumTransporterPhysics.Common.SolverParameters.Gamma2` [get], [set]

Voltage conversion factor

5.26.4.4 `double InvertedPendulumTransporterPhysics.Common.SolverParameters.HorizontalWindForce` [get], [set]

Horizontal wind force

5.26.4.5 `double InvertedPendulumTransporterPhysics.Common.SolverParameters.PendulumLength` [get], [set]

Pendulum Length

5.26.4.6 `double InvertedPendulumTransporterPhysics.Common.SolverParameters.PendulumMass` [get], [set]

Pendulum mass

5.26.4.7 `double InvertedPendulumTransporterPhysics.Common.SolverParameters.VerticalWindForce` [get], [set]

Vertical wind force

5.26.4.8 `double InvertedPendulumTransporterPhysics.Common.SolverParameters.Voltage` [get], [set]

Motor voltage

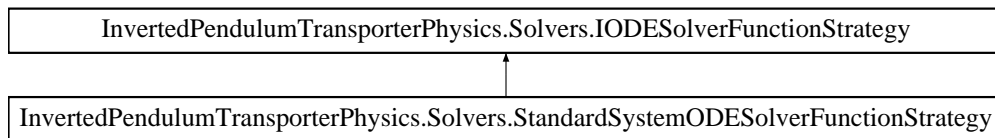
The documentation for this class was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/PhysicsLibrary/Common/SolverParameters.cs

## 5.27 InvertedPendulumTransporterPhysics.Solvers.StandardSystemODESolverFunction↔ Strategy Class Reference

Strategy for base dynamics system

Inheritance diagram for InvertedPendulumTransporterPhysics.Solvers.StandardSystemODESolverFunction↔  
Strategy:



## Public Member Functions

- void [ODESolverFunction](#) (double[] y, double x, double[] dy, object obj)  
*Function of solving the state-space equations in differential form*

### 5.27.1 Detailed Description

Strategy for base dynamics system

### 5.27.2 Member Function Documentation

- 5.27.2.1 void [InvertedPendulumTransporterPhysics.Solvers.StandardSystemODESolverFunctionStrategy.ODESolverFunction](#) (double[] y, double x, double[] dy, object obj) [inline]

Function of solving the state-space equations in differential form

Parameters

y	System state array
x	Time stamps array
dy	System state derivatives array
obj	Solver parameters

Implements [InvertedPendulumTransporterPhysics.Solvers.IODESolverFunctionStrategy](#).

The documentation for this class was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/PhysicsLibrary/Solvers/StandardSystemODESolver↔  
FunctionStrategy.cs

## 5.28 InvertedPendulumTransporterPhysics.Common.SystemState Class Reference

Container for the whole system state

## Public Member Functions

- [SystemState](#) ()  
*Class constructor*
- void [ResetSystemParameters](#) ()  
*Reset system parameters to default*
- void [Reset](#) (double xCoordAngle=0.0, double yCoordAngle=0.0, double xCoordPosition=0.0, double y↔  
CoordPosition=0.0)  
*Reset system state*
- double[] [ToTimeArray](#) ()  
*Convert time to array*
- void [UpdateTimer](#) ()

- Update simulation time*
- void [ResetTimer](#) ()
- Reset simulation time*
- void [UpdateSystemStateX](#) ([OneDimensionalSystemState](#) xState)
- Update X-coordinate subsystem state*
- void [UpdateSystemStateY](#) ([OneDimensionalSystemState](#) yState)
- Update Y-coordinate subsystem state*
- [Point3D](#) [GetSystemPosition](#) ()
- Get actual cart position*

## Public Attributes

- double [DefaultTimeDelta](#) = 0.01
- Default simulation time delta*

## Properties

- [OneDimensionalSystemState](#) [StateX](#) [get]
- X-coordinate subsystem state*
- [OneDimensionalSystemState](#) [StateY](#) [get]
- Y-coordinate subsystem state*
- [OneDimensionalSystemState](#) [LastStateX](#) [get]
- Last X-coordinate subsystem state*
- [OneDimensionalSystemState](#) [LastStateY](#) [get]
- Last Y-coordinate subsystem state*
- double [MaxAngle](#) [get]
- Max pendulum angle*
- double [MinAngle](#) [get]
- Min pendulum angle*
- double [Time](#) [get, set]
- Current simulation time*
- double [TimeDelta](#) [get, set]
- Current simulation time delta*
- [SolverParameters](#) [SolverParameters](#) [get, set]
- Current solver parameters*

### 5.28.1 Detailed Description

Container for the whole system state

### 5.28.2 Constructor & Destructor Documentation

#### 5.28.2.1 [InvertedPendulumTransporterPhysics.Common.SystemState.SystemState](#) ( ) [inline]

Class constructor

### 5.28.3 Member Function Documentation

#### 5.28.3.1 `Point3D InvertedPendulumTransporterPhysics.Common.SystemState.GetSystemPosition ( ) [inline]`

Get actual cart position

Returns

Point with 2-dim position

#### 5.28.3.2 `void InvertedPendulumTransporterPhysics.Common.SystemState.Reset ( double xCoordAngle = 0.0, double yCoordAngle = 0.0, double xCoordPosition = 0.0, double yCoordPosition = 0.0 ) [inline]`

Reset system state

Parameters

<i>xCoordAngle</i>	New pendulum angle in X-coordinate
<i>yCoordAngle</i>	New pendulum angle in Y-coordinate
<i>xCoordPosition</i>	New cart position in X-coordinate
<i>yCoordPosition</i>	New cart position in Y-coordinate

#### 5.28.3.3 `void InvertedPendulumTransporterPhysics.Common.SystemState.ResetSystemParameters ( ) [inline]`

Reset system parameters to default

#### 5.28.3.4 `void InvertedPendulumTransporterPhysics.Common.SystemState.ResetTimer ( ) [inline]`

Reset simulation time

#### 5.28.3.5 `double [] InvertedPendulumTransporterPhysics.Common.SystemState.ToTimeArray ( ) [inline]`

Convert time to array

Returns

Array of time stamps

#### 5.28.3.6 `void InvertedPendulumTransporterPhysics.Common.SystemState.UpdateSystemStateX ( OneDimensionalSystemState xState ) [inline]`

Update X-coordinate subsystem state

Parameters

<i>xState</i>	New subsystem state
---------------	---------------------

#### 5.28.3.7 `void InvertedPendulumTransporterPhysics.Common.SystemState.UpdateSystemStateY ( OneDimensionalSystemState yState ) [inline]`

Update Y-coordinate subsystem state

## Parameters

<i>xState</i>	New subsystem state
---------------	---------------------

5.28.3.8 `void InvertedPendulumTransporterPhysics.Common.SystemState.UpdateTimer ( ) [inline]`

Update simulation time

## 5.28.4 Member Data Documentation

5.28.4.1 `double InvertedPendulumTransporterPhysics.Common.SystemState.DefaultTimeDelta = 0.01`

Default simulation time delta

## 5.28.5 Property Documentation

5.28.5.1 `OneDimensionalSystemState InvertedPendulumTransporterPhysics.Common.SystemState.LastStateX [get]`

Last X-coordinate subsystem state

5.28.5.2 `OneDimensionalSystemState InvertedPendulumTransporterPhysics.Common.SystemState.LastStateY [get]`

Last Y-coordinate subsystem state

5.28.5.3 `double InvertedPendulumTransporterPhysics.Common.SystemState.MaxAngle [get]`

Max pendulum angle

5.28.5.4 `double InvertedPendulumTransporterPhysics.Common.SystemState.MinAngle [get]`

Min pendulum angle

5.28.5.5 `SolverParameters InvertedPendulumTransporterPhysics.Common.SystemState.SolverParameters [get], [set]`

Current solver parameters

5.28.5.6 `OneDimensionalSystemState InvertedPendulumTransporterPhysics.Common.SystemState.StateX [get]`

X-coordinate subsystem state

5.28.5.7 `OneDimensionalSystemState InvertedPendulumTransporterPhysics.Common.SystemState.StateY [get]`

Y-coordinate subsystem state

5.28.5.8 `double InvertedPendulumTransporterPhysics.Common.SystemState.Time [get], [set]`

Current simulation time

5.28.5.9 double InvertedPendulumTransporterPhysics.Common.SystemState.TimeDelta [get], [set]

Current simulation time delta

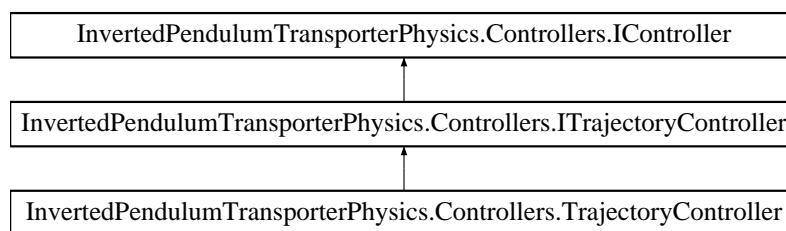
The documentation for this class was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/PhysicsLibrary/Common/SystemState.cs

## 5.29 InvertedPendulumTransporterPhysics.Controllers.TrajectoryController Class Reference

Controller for trajectory tracking

Inheritance diagram for InvertedPendulumTransporterPhysics.Controllers.TrajectoryController:



### Public Member Functions

- [TrajectoryController](#) ()  
*Class constructor*
- void [Reset](#) ()  
*Reset controller to default state*
- void [Clear](#) ()  
*Clear trajectory*
- void [SetAccuracy](#) ([AccuracyType](#) accuracy)  
*Set trajectory tracking accuracy*
- Point3DCollection [LoadTrajectory](#) (string fileName=null)  
*Load trajectory from file*
- string [SaveTrajectory](#) (List< Point3D > trajectory)  
*Save trajectory to file*
- Point3D [GetTargetStartPosition](#) ()  
*Get trajectory beginning position*
- Point3D [GetTargetPosition](#) (double x, double y, out bool nextCheckPoint)  
*Get actual target control point*
- Point3D [GetTargetSmoothPosition](#) (double x, double y)  
*Get target position as a smooth combination of control points [the function is deprecated]*
- Point3D [GetTargetApproximatePosition](#) (double x, double y)  
*Get target position as a approximation between control points [the function is deprecated]*

### Properties

- double **AverageDistance** [get]
- bool **TrajectoryAchieved** [get]
- bool **TrajectoryEnabled** [get]

### 5.29.1 Detailed Description

Controller for trajectory tracking

### 5.29.2 Constructor & Destructor Documentation

#### 5.29.2.1 InvertedPendulumTransporterPhysics.Controllers.TrajectoryController.TrajectoryController ( ) [inline]

Class constructor

### 5.29.3 Member Function Documentation

#### 5.29.3.1 void InvertedPendulumTransporterPhysics.Controllers.TrajectoryController.Clear ( ) [inline]

Clear trajectory

Implements [InvertedPendulumTransporterPhysics.Controllers.ITrajectoryController](#).

#### 5.29.3.2 Point3D InvertedPendulumTransporterPhysics.Controllers.TrajectoryController.GetTargetApproximatePosition ( double x, double y ) [inline]

Get target position as a approximation between control points [the function is deprecated]

Parameters

<i>x</i>	Cart position in X-Coordinate
<i>y</i>	Cart position in Y-Coordinate

Returns

Target position

#### 5.29.3.3 Point3D InvertedPendulumTransporterPhysics.Controllers.TrajectoryController.GetTargetPosition ( double x, double y, out bool *nextCheckPoint* ) [inline]

Get actual target control point

Parameters

<i>x</i>	Cart position in X-coordinate
<i>y</i>	Cart position in Y-coordinate
<i>nextCheckPoint</i>	Info about target point change

Returns

Target control point

Implements [InvertedPendulumTransporterPhysics.Controllers.ITrajectoryController](#).

#### 5.29.3.4 Point3D InvertedPendulumTransporterPhysics.Controllers.TrajectoryController.GetTargetSmoothPosition ( double x, double y ) [inline]

Get target position as a smooth combination of control points [the function is deprecated]



## Parameters

<i>x</i>	Cart position in X-Coordinate
<i>y</i>	Cart position in Y-Coordinate

## Returns

Target position

**5.29.3.5** `Point3D InvertedPendulumTransporterPhysics.Controllers.TrajectoryController.GetTargetStartPosition ( )`  
[inline]

Get trajectory beginning position

## Returns

Implements [InvertedPendulumTransporterPhysics.Controllers.ITrajectoryController](#).

**5.29.3.6** `Point3DCollection InvertedPendulumTransporterPhysics.Controllers.TrajectoryController.LoadTrajectory ( string  
fileName = null )` [inline]

Load trajectory from file

## Parameters

<i>fileName</i>	File full path (if null additionally open file browser)
-----------------	---

## Returns

Control points collection with doubled points for trajectory visualization

Implements [InvertedPendulumTransporterPhysics.Controllers.ITrajectoryController](#).

**5.29.3.7** `void InvertedPendulumTransporterPhysics.Controllers.TrajectoryController.Reset ( )` [inline]

Reset controller to default state

Implements [InvertedPendulumTransporterPhysics.Controllers.IController](#).

**5.29.3.8** `string InvertedPendulumTransporterPhysics.Controllers.TrajectoryController.SaveTrajectory ( List< Point3D >  
trajectory )` [inline]

Save trajectory to file

## Parameters

<i>trajectory</i>	List of control points
-------------------	------------------------

## Returns

File full path

Implements [InvertedPendulumTransporterPhysics.Controllers.ITrajectoryController](#).

5.29.3.9 `void InvertedPendulumTransporterPhysics.Controllers.TrajectoryController.SetAccuracy ( AccuracyType accuracy )`  
`[inline]`

Set trajectory tracking accuracy

## Parameters

<i>accuracy</i>	Accuracy type
-----------------	---------------

Implements [InvertedPendulumTransporterPhysics.Controllers.ITrajectoryController](#).

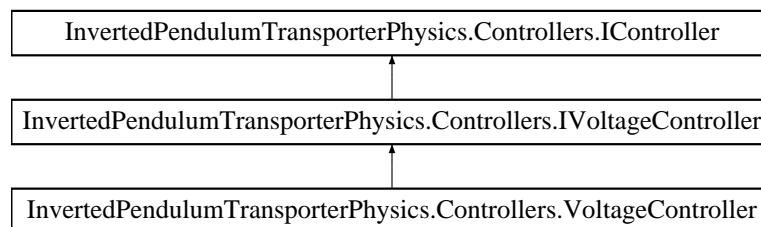
The documentation for this class was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/PhysicsLibrary/Controllers/TrajectoryController.cs

## 5.30 InvertedPendulumTransporterPhysics.Controllers.VoltageController Class Reference

Controller for motor voltage

Inheritance diagram for InvertedPendulumTransporterPhysics.Controllers.VoltageController:



### Public Member Functions

- [VoltageController](#) ()  
*Class constructor*
- double [GetVoltage](#) ()  
*Get result voltage from regulation process*
- void [SetTime](#) (double time)  
*Set current simulation time (for sinusoidal methods)*
- void [Reset](#) ()  
*Reset controller to default state*
- void [Reset](#) (double timeDelta)  
*Reset controller with given time delta*
- void [SetControlError](#) (double angleError, double positionError)  
*Set control error*
- void [SetUserAngle](#) (double userAngle)  
*Setup angle generated by user (for none control)*

### Public Attributes

- const [ControlType](#) **DefaultControlType** = ControlType.PID

### Properties

- [ControlType](#) **ControlType** [get, set]

### 5.30.1 Detailed Description

Controller for motor voltage

### 5.30.2 Constructor & Destructor Documentation

5.30.2.1 `InvertedPendulumTransporterPhysics.Controllers.VoltageController.VoltageController ( )` `[inline]`

Class constructor

### 5.30.3 Member Function Documentation

5.30.3.1 `double InvertedPendulumTransporterPhysics.Controllers.VoltageController.GetVoltage ( )` `[inline]`

Get result voltage from regulation process

Returns

Motor Voltage

Implements [InvertedPendulumTransporterPhysics.Controllers.IVoltageController](#).

5.30.3.2 `void InvertedPendulumTransporterPhysics.Controllers.VoltageController.Reset ( )` `[inline]`

Reset controller to default state

Implements [InvertedPendulumTransporterPhysics.Controllers.IController](#).

5.30.3.3 `void InvertedPendulumTransporterPhysics.Controllers.VoltageController.Reset ( double timeDelta )` `[inline]`

Reset controller with given time delta

Parameters

<i>timeDelta</i>	Actual time delta
------------------	-------------------

Implements [InvertedPendulumTransporterPhysics.Controllers.IVoltageController](#).

5.30.3.4 `void InvertedPendulumTransporterPhysics.Controllers.VoltageController.SetControlError ( double angleError, double positionError )` `[inline]`

Set control error

Parameters

<i>angleError</i>	Difference between desired angle and actual one
<i>positionError</i>	Difference between desired position and actual one

Implements [InvertedPendulumTransporterPhysics.Controllers.IVoltageController](#).

5.30.3.5 `void InvertedPendulumTransporterPhysics.Controllers.VoltageController.SetTime ( double time )` `[inline]`

Set current simulation time (for sinusoidal methods)

## Parameters

<i>time</i>	Current simuation time
-------------	------------------------

Implements [InvertedPendulumTransporterPhysics.Controllers.IVoltageController](#).

5.30.3.6 `void InvertedPendulumTransporterPhysics.Controllers.VoltageController.SetUserAngle ( double userAngle )`  
`[inline]`

Setup angle generated by user (for none control)

## Parameters

<i>userAngle</i>	User angle value
------------------	------------------

Implements [InvertedPendulumTransporterPhysics.Controllers.IVoltageController](#).

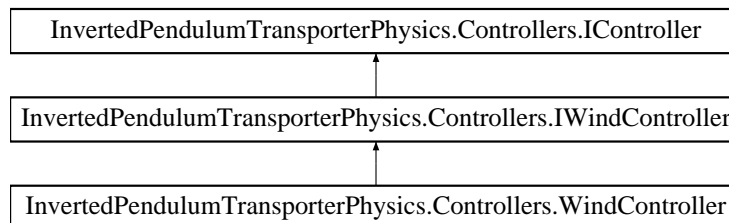
The documentation for this class was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/PhysicsLibrary/Controllers/VoltageController.cs

## 5.31 InvertedPendulumTransporterPhysics.Controllers.WindController Class Reference

Controller for wind power

Inheritance diagram for InvertedPendulumTransporterPhysics.Controllers.WindController:



### Public Member Functions

- [WindController](#) ()  
*Class constructor*
- double [GetXCoordWindPower](#) ()  
*Get final wind power in X-coordinate*
- double [GetYCoordWindPower](#) ()  
*Get final wind power in X-coordinate*
- double [GetZCoordWindPower](#) ()  
*Get final wind power in X-coordinate*
- Vector3D [UpdateWindForce](#) ()  
*Update wind force*
- void [Reset](#) ()  
*Reset controller to default state*

### Public Attributes

- const double [DefaultWindChangeSpeed](#) = 0.5  
*Default wind change speed*

- const [WindType](#) **DefaultWindType** = WindType.RandomSmooth  
*Default wind type*

## Properties

- [WindType](#) **WindType** [get, set]
- double **WindPower** [get, set]
- double **WindChangeSpeed** [get, set]
- double **MaxWindPower** [get]
- double **MinWindPower** [get]
- double **DefaultWindPower** [get]

### 5.31.1 Detailed Description

Controller for wind power

### 5.31.2 Constructor & Destructor Documentation

5.31.2.1 [InvertedPendulumTransporterPhysics.Controllers.WindController](#).[WindController](#) ( ) [inline]

Class constructor

### 5.31.3 Member Function Documentation

5.31.3.1 double [InvertedPendulumTransporterPhysics.Controllers.WindController](#).[GetXCoordWindPower](#) ( ) [inline]

Get final wind power in X-coordinate

Returns

Wind power

Implements [InvertedPendulumTransporterPhysics.Controllers.IWindController](#).

5.31.3.2 double [InvertedPendulumTransporterPhysics.Controllers.WindController](#).[GetYCoordWindPower](#) ( ) [inline]

Get final wind power in X-coordinate

Returns

Wind power

Implements [InvertedPendulumTransporterPhysics.Controllers.IWindController](#).

5.31.3.3 double [InvertedPendulumTransporterPhysics.Controllers.WindController](#).[GetZCoordWindPower](#) ( ) [inline]

Get final wind power in X-coordinate

Returns

Wind power

Implements [InvertedPendulumTransporterPhysics.Controllers.IWindController](#).

5.31.3.4 `void InvertedPendulumTransporterPhysics.Controllers.WindController.Reset ( ) [inline]`

Reset controller to default state

Implements [InvertedPendulumTransporterPhysics.Controllers.IController](#).

5.31.3.5 `Vector3D InvertedPendulumTransporterPhysics.Controllers.WindController.UpdateWindForce ( ) [inline]`

Update wind force

Returns

Wind force direction

Implements [InvertedPendulumTransporterPhysics.Controllers.IWindController](#).

## 5.31.4 Member Data Documentation

5.31.4.1 `const double InvertedPendulumTransporterPhysics.Controllers.WindController.DefaultWindChangeSpeed = 0.5`

Default wind change speed

5.31.4.2 `const WindType InvertedPendulumTransporterPhysics.Controllers.WindController.DefaultWindType = WindType.RandomSmooth`

Default wind type

The documentation for this class was generated from the following file:

- C:/Users/Arkady/Downloads/Murzyński Backup/BIB/PhysicsLibrary/Controllers/WindController.cs





# Index

AboutWindow  
    InvertedPendulumTransporter::Windows::About↔  
        Window, [11](#)

AccuracyType  
    InvertedPendulumTransporterPhysics::Controllers,  
        [10](#)

Angle  
    InvertedPendulumTransporterPhysics::Common↔  
        ::OneDimensionalSystemState, [41](#)

AngleErrorPointsX  
    InvertedPendulumTransporter::Controls::Plots↔  
        Control, [47](#)

AngleErrorPointsY  
    InvertedPendulumTransporter::Controls::Plots↔  
        Control, [47](#)

AngularVelocity  
    InvertedPendulumTransporterPhysics::Common↔  
        ::OneDimensionalSystemState, [41](#)

AverageDistance  
    InvertedPendulumTransporterPhysics::Controllers↔  
        ::ITrajectoryController, [31](#)

CalculateAnglePIDCorrection  
    InvertedPendulumTransporterPhysics::Controllers↔  
        ::PIDCorrector, [44](#)

CalculateParallelPositionAnglePIDCorrection  
    InvertedPendulumTransporterPhysics::Controllers↔  
        ::PIDCorrector, [44](#)

CalculatePositionPIDCorrection  
    InvertedPendulumTransporterPhysics::Controllers↔  
        ::PIDCorrector, [44](#)

Cart  
    InvertedPendulumTransporter::Models::Cart, [13](#)

cart  
    InvertedPendulumTransporter::Controls::Scene↔  
        Control, [51](#)

CartLinkPoint  
    InvertedPendulumTransporter::Models::IPendulum,  
        [26](#)

CartMass  
    InvertedPendulumTransporter::Windows::Main↔  
        Window, [37](#)  
    InvertedPendulumTransporterPhysics::Common↔  
        ::SolverParameters, [54](#)

CartPositionX  
    InvertedPendulumTransporter::Windows::Main↔  
        Window, [37](#)

CartPositionY  
    InvertedPendulumTransporter::Windows::Main↔  
        Window, [37](#)

CartVelocityX  
    InvertedPendulumTransporter::Windows::Main↔  
        Window, [37](#)

CartVelocityY  
    InvertedPendulumTransporter::Windows::Main↔  
        Window, [37](#)

Clear  
    InvertedPendulumTransporterPhysics::Controllers↔  
        ::ITrajectoryController, [28](#)  
    InvertedPendulumTransporterPhysics::Controllers↔  
        ::TrajectoryController, [60](#)

ClearTrajectory  
    InvertedPendulumTransporter::Controls::Scene↔  
        Control, [49](#)

ControlType  
    InvertedPendulumTransporterPhysics::Controllers,  
        [10](#)  
    InvertedPendulumTransporterPhysics::Controllers↔  
        ::IVoltageController, [33](#)

CreateTrajectoryWindow  
    InvertedPendulumTransporter::Windows::Create↔  
        TrajectoryWindow, [15](#)

DefaultTimeDelta  
    InvertedPendulumTransporterPhysics::Common↔  
        ::SystemState, [58](#)

DefaultWindChangeSpeed  
    InvertedPendulumTransporterPhysics::Controllers↔  
        ::WindController, [67](#)

DefaultWindPower  
    InvertedPendulumTransporterPhysics::Controllers↔  
        ::IWindController, [34](#)

DefaultWindType  
    InvertedPendulumTransporterPhysics::Controllers↔  
        ::WindController, [67](#)

G  
    InvertedPendulumTransporterPhysics::Common↔  
        ::SolverParameters, [54](#)

GameController  
    InvertedPendulumTransporterPhysics::Controllers↔  
        ::GameController, [17](#)

GameEnabled  
    InvertedPendulumTransporterPhysics::Controllers↔  
        ::IGameController, [20](#)

GamePlaying  
    InvertedPendulumTransporterPhysics::Controllers↔  
        ::IGameController, [20](#)

Gamma1

- InvertedPendulumTransporterPhysics::Common↔  
::SolverParameters, [54](#)
- Gamma2
  - InvertedPendulumTransporterPhysics::Common↔  
::SolverParameters, [54](#)
- GetSystemPosition
  - InvertedPendulumTransporterPhysics::Common↔  
::SystemState, [57](#)
- GetTargetApproximatePosition
  - InvertedPendulumTransporterPhysics::Controllers↔  
::TrajectoryController, [60](#)
- GetTargetPosition
  - InvertedPendulumTransporterPhysics::Controllers↔  
::ITrajectoryController, [28](#)
  - InvertedPendulumTransporterPhysics::Controllers↔  
::TrajectoryController, [60](#)
- GetTargetSmoothPosition
  - InvertedPendulumTransporterPhysics::Controllers↔  
::TrajectoryController, [60](#)
- GetTargetStartPosition
  - InvertedPendulumTransporterPhysics::Controllers↔  
::ITrajectoryController, [30](#)
  - InvertedPendulumTransporterPhysics::Controllers↔  
::TrajectoryController, [61](#)
- GetVoltage
  - InvertedPendulumTransporterPhysics::Controllers↔  
::IVoltageController, [32](#)
  - InvertedPendulumTransporterPhysics::Controllers↔  
::VoltageController, [64](#)
- GetXCoordWindPower
  - InvertedPendulumTransporterPhysics::Controllers↔  
::IWindController, [34](#)
  - InvertedPendulumTransporterPhysics::Controllers↔  
::WindController, [66](#)
- GetYCoordWindPower
  - InvertedPendulumTransporterPhysics::Controllers↔  
::IWindController, [34](#)
  - InvertedPendulumTransporterPhysics::Controllers↔  
::WindController, [66](#)
- GetZCoordWindPower
  - InvertedPendulumTransporterPhysics::Controllers↔  
::IWindController, [34](#)
  - InvertedPendulumTransporterPhysics::Controllers↔  
::WindController, [66](#)
- HandleKey
  - InvertedPendulumTransporterPhysics::Controllers↔  
::GameController, [18](#)
  - InvertedPendulumTransporterPhysics::Controllers↔  
::IGameController, [20](#)
- HorizontalWindForce
  - InvertedPendulumTransporterPhysics::Common↔  
::SolverParameters, [54](#)
- Initialize
  - InvertedPendulumTransporter::Models::Cart, [13](#)
  - InvertedPendulumTransporter::Models::IModel, [21](#)
  - InvertedPendulumTransporter::Models::Pendulum,  
[43](#)
- InvertedPendulumTransporter::Models::Simulation↔  
Area, [52](#)
- InitializeDefault
  - InvertedPendulumTransporterPhysics::Common↔  
::OneDimensionalSystemState, [41](#)
- InvertedPendulumTransporter, [7](#)
- InvertedPendulumTransporter.App, [12](#)
- InvertedPendulumTransporter.Controls, [7](#)
- InvertedPendulumTransporter.Controls.PlotsControl, [45](#)
- InvertedPendulumTransporter.Controls.SceneControl,  
[48](#)
- InvertedPendulumTransporter.Models, [7](#)
- InvertedPendulumTransporter.Models.Cart, [12](#)
- InvertedPendulumTransporter.Models.ICart, [18](#)
- InvertedPendulumTransporter.Models.IModel, [21](#)
- InvertedPendulumTransporter.Models.IPendulum, [26](#)
- InvertedPendulumTransporter.Models.ISimulationArea,  
[27](#)
- InvertedPendulumTransporter.Models.Pendulum, [42](#)
- InvertedPendulumTransporter.Models.SimulationArea,  
[51](#)
- InvertedPendulumTransporter.Properties, [8](#)
- InvertedPendulumTransporter.Windows, [8](#)
- InvertedPendulumTransporter.Windows.AboutWindow,  
[11](#)
- InvertedPendulumTransporter.Windows.Create↔  
TrajectoryWindow, [15](#)
- InvertedPendulumTransporter.Windows.MainWindow,  
[35](#)
- InvertedPendulumTransporter::Controls::PlotsControl
  - AngleErrorPointsX, [47](#)
  - AngleErrorPointsY, [47](#)
  - PassParameters, [46](#)
  - PlotsControl, [46](#)
  - PositionErrorPointsX, [47](#)
  - PositionErrorPointsY, [47](#)
  - ResetPlots, [46](#)
  - UpdateAngleErrorPlots, [46](#)
  - UpdatePositionErrorPlots, [47](#)
  - UpdateVoltagePlots, [47](#)
  - VoltagePointsX, [47](#)
  - VoltagePointsY, [47](#)
- InvertedPendulumTransporter::Controls::SceneControl
  - cart, [51](#)
  - ClearTrajectory, [49](#)
  - pendulum, [51](#)
  - ResetSimulation, [49](#)
  - SceneControl, [49](#)
  - SetupHighLevelGraphics, [49](#)
  - SetupLowLevelGraphics, [49](#)
  - ShowCartTrajectory, [49](#)
  - ShowPendulumTrajectory, [50](#)
  - ShowTargetTrajectory, [50](#)
  - simulationArea, [51](#)
  - UpdateCamera, [50](#)
  - UpdateFrame, [50](#)
  - UpdateState, [50](#)
  - UpdateTrajectory, [50](#)

- UpdateWindDirection, 51
- InvertedPendulumTransporter::Models::Cart
  - Cart, 13
  - Initialize, 13
  - platformHeightFactor, 15
  - SetupHighLevelGraphics, 13
  - SetupLowLevelGraphics, 13
  - UpdateState, 13
  - wheelRadius, 15
- InvertedPendulumTransporter::Models::ICart
  - PlatformSize, 19
- InvertedPendulumTransporter::Models::IModel
  - Initialize, 21
  - Model, 22
  - SetupHighLevelGraphics, 21
  - SetupLowLevelGraphics, 22
  - UpdateState, 22
- InvertedPendulumTransporter::Models::IPendulum
  - CartLinkPoint, 26
  - MassLinkPoint, 27
  - RodLength, 27
- InvertedPendulumTransporter::Models::ISimulationArea
  - Size, 27
- InvertedPendulumTransporter::Models::Pendulum
  - Initialize, 43
  - Pendulum, 42
  - SetupHighLevelGraphics, 43
  - SetupLowLevelGraphics, 43
  - UpdateState, 43
- InvertedPendulumTransporter::Models::SimulationArea
  - Initialize, 52
  - SetupHighLevelGraphics, 52
  - SetupLowLevelGraphics, 52
  - SimulationArea, 52
  - UpdateState, 52
- InvertedPendulumTransporter::Windows
  - WindowType, 8
- InvertedPendulumTransporter::Windows::AboutWindow
  - AboutWindow, 11
  - SetupHelpWindow, 11
  - SetupWindowType, 12
- InvertedPendulumTransporter::Windows::Create↔
  - TrajectoryWindow
  - CreateTrajectoryWindow, 15
  - TrajectoryLoaded, 16
  - TrajectoryPoints, 16
- InvertedPendulumTransporter::Windows::MainWindow
  - CartMass, 37
  - CartPositionX, 37
  - CartPositionY, 37
  - CartVelocityX, 37
  - CartVelocityY, 37
  - MainWindow, 36
  - MaxAngle, 37
  - MaxWindPower, 37
  - MinAngle, 37
  - MinWindPower, 37
  - PendulumAngleX, 37
  - PendulumAngleY, 37
  - PendulumAngularVelocityX, 37
  - PendulumAngularVelocityY, 37
  - PendulumMass, 38
  - PropertyChanged, 38
  - RodLength, 38
  - TimeDelta, 38
  - WindChangeSpeed, 38
  - WindPower, 38
  - XCoordAngle, 38
  - YCoordAngle, 38
- InvertedPendulumTransporterPhysics, 8
- InvertedPendulumTransporterPhysics.Common, 9
- InvertedPendulumTransporterPhysics.Common.One↔
  - DimensionalSystemState, 40
- InvertedPendulumTransporterPhysics.Common.↔
  - SolverParameters, 53
- InvertedPendulumTransporterPhysics.Common.↔
  - SystemState, 55
- InvertedPendulumTransporterPhysics.Controllers, 9
- InvertedPendulumTransporterPhysics.Controllers.↔
  - GameController, 16
- InvertedPendulumTransporterPhysics.Controllers.I↔
  - Controller, 19
- InvertedPendulumTransporterPhysics.Controllers.I↔
  - GameController, 19
- InvertedPendulumTransporterPhysics.Controllers.I↔
  - TrajectoryController, 28
- InvertedPendulumTransporterPhysics.Controllers.I↔
  - VoltageController, 31
- InvertedPendulumTransporterPhysics.Controllers.I↔
  - WindController, 33
- InvertedPendulumTransporterPhysics.Controllers.PID↔
  - Corrector, 43
- InvertedPendulumTransporterPhysics.Controllers.↔
  - TrajectoryController, 59
- InvertedPendulumTransporterPhysics.Controllers.↔
  - VoltageController, 63
- InvertedPendulumTransporterPhysics.Controllers.↔
  - WindController, 65
- InvertedPendulumTransporterPhysics.Solvers, 10
- InvertedPendulumTransporterPhysics.Solvers.IODE↔
  - Solver, 23
- InvertedPendulumTransporterPhysics.Solvers.IODE↔
  - SolverFunctionStrategy, 25
- InvertedPendulumTransporterPhysics.Solvers.Interfered↔
  - SystemODESolverFunctionStrategy, 22
- InvertedPendulumTransporterPhysics.Solvers.ODE↔
  - Solver, 38
- InvertedPendulumTransporterPhysics.Solvers.Standard↔
  - SystemODESolverFunctionStrategy, 54
- InvertedPendulumTransporterPhysics::Common::One↔
  - DimensionalSystemState
  - Angle, 41
  - AngularVelocity, 41
  - InitializeDefault, 41
  - OneDimensionalSystemState, 41
  - Position, 41

- ToStateArray, [41](#)
- Velocity, [41](#)
- InvertedPendulumTransporterPhysics::Common::↔
  - SolverParameters
  - CartMass, [54](#)
  - G, [54](#)
  - Gamma1, [54](#)
  - Gamma2, [54](#)
  - HorizontalWindForce, [54](#)
  - PendulumLength, [54](#)
  - PendulumMass, [54](#)
  - SolverParameters, [53](#)
  - VerticalWindForce, [54](#)
  - Voltage, [54](#)
- InvertedPendulumTransporterPhysics::Common::↔
  - SystemState
  - DefaultTimeDelta, [58](#)
  - GetSystemPosition, [57](#)
  - LastStateX, [58](#)
  - LastStateY, [58](#)
  - MaxAngle, [58](#)
  - MinAngle, [58](#)
  - Reset, [57](#)
  - ResetSystemParameters, [57](#)
  - ResetTimer, [57](#)
  - SolverParameters, [58](#)
  - StateX, [58](#)
  - StateY, [58](#)
  - SystemState, [56](#)
  - Time, [58](#)
  - TimeDelta, [58](#)
  - ToTimeArray, [57](#)
  - UpdateSystemStateX, [57](#)
  - UpdateSystemStateY, [57](#)
  - UpdateTimer, [58](#)
- InvertedPendulumTransporterPhysics::Controllers
  - AccuracyType, [10](#)
  - ControlType, [10](#)
  - WindType, [10](#)
- InvertedPendulumTransporterPhysics::Controllers::↔
  - GameController
  - GameController, [17](#)
  - HandleKey, [18](#)
  - Reset, [18](#)
- InvertedPendulumTransporterPhysics::Controllers::!↔
  - Controller
  - Reset, [19](#)
- InvertedPendulumTransporterPhysics::Controllers::!↔
  - GameController
  - GameEnabled, [20](#)
  - GamePlaying, [20](#)
  - HandleKey, [20](#)
  - UserAngleX, [20](#)
  - UserAngleY, [21](#)
- InvertedPendulumTransporterPhysics::Controllers::!↔
  - TrajectoryController
  - AverageDistance, [31](#)
  - Clear, [28](#)
  - GetTargetPosition, [28](#)
  - GetTargetStartPosition, [30](#)
  - LoadTrajectory, [30](#)
  - SaveTrajectory, [30](#)
  - SetAccuracy, [30](#)
  - TrajectoryAchieved, [31](#)
  - TrajectoryEnabled, [31](#)
- InvertedPendulumTransporterPhysics::Controllers::!↔
  - VoltageController
  - ControlType, [33](#)
  - GetVoltage, [32](#)
  - Reset, [32](#)
  - SetControlError, [32](#)
  - SetTime, [32](#)
  - SetUserAngle, [32](#)
- InvertedPendulumTransporterPhysics::Controllers::!↔
  - WindController
  - DefaultWindPower, [34](#)
  - GetXCoordWindPower, [34](#)
  - GetYCoordWindPower, [34](#)
  - GetZCoordWindPower, [34](#)
  - MaxWindPower, [34](#)
  - MinWindPower, [35](#)
  - UpdateWindForce, [34](#)
  - WindChangeSpeed, [35](#)
  - WindPower, [35](#)
  - WindType, [35](#)
- InvertedPendulumTransporterPhysics::Controllers::PI↔
  - DCorrector
  - CalculateAnglePIDCorrection, [44](#)
  - CalculateParallelPositionAnglePIDCorrection, [44](#)
  - CalculatePositionPIDCorrection, [44](#)
  - PIDCorrector, [44](#)
  - Reset, [44](#)
  - SetAngleError, [45](#)
  - SetPositionError, [45](#)
- InvertedPendulumTransporterPhysics::Controllers::↔
  - TrajectoryController
  - Clear, [60](#)
  - GetTargetApproximatePosition, [60](#)
  - GetTargetPosition, [60](#)
  - GetTargetSmoothPosition, [60](#)
  - GetTargetStartPosition, [61](#)
  - LoadTrajectory, [61](#)
  - Reset, [61](#)
  - SaveTrajectory, [61](#)
  - SetAccuracy, [61](#)
  - TrajectoryController, [60](#)
- InvertedPendulumTransporterPhysics::Controllers::↔
  - VoltageController
  - GetVoltage, [64](#)
  - Reset, [64](#)
  - SetControlError, [64](#)
  - SetTime, [64](#)
  - SetUserAngle, [65](#)
  - VoltageController, [64](#)
- InvertedPendulumTransporterPhysics::Controllers::↔
  - WindController

- DefaultWindChangeSpeed, [67](#)
- DefaultWindType, [67](#)
- GetXCoordWindPower, [66](#)
- GetYCoordWindPower, [66](#)
- GetZCoordWindPower, [66](#)
- Reset, [66](#)
- UpdateWindForce, [67](#)
- WindController, [66](#)
- InvertedPendulumTransporterPhysics::Solvers::IODE↔
  - Solver
  - SetupStrategy, [23](#)
  - SolveODESystem, [25](#)
  - UpdateSystemParameters, [25](#)
- InvertedPendulumTransporterPhysics::Solvers::IODE↔
  - SolverFunctionStrategy
  - ODESolverFunction, [26](#)
- InvertedPendulumTransporterPhysics::Solvers::↔
  - InterferedSystemODESolverFunctionStrategy
  - ODESolverFunction, [23](#)
- InvertedPendulumTransporterPhysics::Solvers::ODE↔
  - Solver
  - ODESolver, [39](#)
  - SetupStrategy, [39](#)
  - SolveODESystem, [39](#)
  - UpdateSystemParameters, [40](#)
- InvertedPendulumTransporterPhysics::Solvers::↔
  - StandardSystemODESolverFunctionStrategy
  - ODESolverFunction, [55](#)
- LastStateX
  - InvertedPendulumTransporterPhysics::Common↔
    - ::SystemState, [58](#)
- LastStateY
  - InvertedPendulumTransporterPhysics::Common↔
    - ::SystemState, [58](#)
- LoadTrajectory
  - InvertedPendulumTransporterPhysics::Controllers↔
    - ::ITrajectoryController, [30](#)
  - InvertedPendulumTransporterPhysics::Controllers↔
    - ::TrajectoryController, [61](#)
- MainWindow
  - InvertedPendulumTransporter::Windows::Main↔
    - Window, [36](#)
- MassLinkPoint
  - InvertedPendulumTransporter::Models::IPendulum, [27](#)
- MaxAngle
  - InvertedPendulumTransporter::Windows::Main↔
    - Window, [37](#)
  - InvertedPendulumTransporterPhysics::Common↔
    - ::SystemState, [58](#)
- MaxWindPower
  - InvertedPendulumTransporter::Windows::Main↔
    - Window, [37](#)
  - InvertedPendulumTransporterPhysics::Controllers↔
    - ::IWindController, [34](#)
- MinAngle
  - InvertedPendulumTransporter::Windows::Main↔
    - Window, [37](#)
  - InvertedPendulumTransporterPhysics::Common↔
    - ::SystemState, [58](#)
- MinWindPower
  - InvertedPendulumTransporter::Windows::Main↔
    - Window, [37](#)
  - InvertedPendulumTransporterPhysics::Controllers↔
    - ::IWindController, [35](#)
- Model
  - InvertedPendulumTransporter::Models::IModel, [22](#)
- ODESolver
  - InvertedPendulumTransporterPhysics::Solvers::↔
    - ODESolver, [39](#)
- ODESolverFunction
  - InvertedPendulumTransporterPhysics::Solvers::I↔
    - ODESolverFunctionStrategy, [26](#)
  - InvertedPendulumTransporterPhysics::Solvers↔
    - ::InterferedSystemODESolverFunction↔
      - Strategy, [23](#)
    - InvertedPendulumTransporterPhysics::Solvers::↔
      - StandardSystemODESolverFunctionStrategy, [55](#)
- OneDimensionalSystemState
  - InvertedPendulumTransporterPhysics::Common↔
    - ::OneDimensionalSystemState, [41](#)
- PIDCorrector
  - InvertedPendulumTransporterPhysics::Controllers↔
    - ::PIDCorrector, [44](#)
- PassParameters
  - InvertedPendulumTransporter::Controls::Plots↔
    - Control, [46](#)
- Pendulum
  - InvertedPendulumTransporter::Models::Pendulum, [42](#)
- pendulum
  - InvertedPendulumTransporter::Controls::Scene↔
    - Control, [51](#)
- PendulumAngleX
  - InvertedPendulumTransporter::Windows::Main↔
    - Window, [37](#)
- PendulumAngleY
  - InvertedPendulumTransporter::Windows::Main↔
    - Window, [37](#)
- PendulumAngularVelocityX
  - InvertedPendulumTransporter::Windows::Main↔
    - Window, [37](#)
- PendulumAngularVelocityY
  - InvertedPendulumTransporter::Windows::Main↔
    - Window, [37](#)
- PendulumLength
  - InvertedPendulumTransporterPhysics::Common↔
    - ::SolverParameters, [54](#)
- PendulumMass
  - InvertedPendulumTransporter::Windows::Main↔
    - Window, [38](#)

- InvertedPendulumTransporterPhysics::Common↔  
::SolverParameters, 54
- platformHeightFactor
  - InvertedPendulumTransporter::Models::Cart, 15
- PlatformSize
  - InvertedPendulumTransporter::Models::ICart, 19
- PlotsControl
  - InvertedPendulumTransporter::Controls::Plots↔  
Control, 46
- Position
  - InvertedPendulumTransporterPhysics::Common↔  
::OneDimensionalSystemState, 41
- PositionErrorPointsX
  - InvertedPendulumTransporter::Controls::Plots↔  
Control, 47
- PositionErrorPointsY
  - InvertedPendulumTransporter::Controls::Plots↔  
Control, 47
- PropertyChanged
  - InvertedPendulumTransporter::Windows::Main↔  
Window, 38
- Reset
  - InvertedPendulumTransporterPhysics::Common↔  
::SystemState, 57
  - InvertedPendulumTransporterPhysics::Controllers↔  
::GameController, 18
  - InvertedPendulumTransporterPhysics::Controllers↔  
::IController, 19
  - InvertedPendulumTransporterPhysics::Controllers↔  
::IVoltageController, 32
  - InvertedPendulumTransporterPhysics::Controllers↔  
::PIDCorrector, 44
  - InvertedPendulumTransporterPhysics::Controllers↔  
::TrajectoryController, 61
  - InvertedPendulumTransporterPhysics::Controllers↔  
::VoltageController, 64
  - InvertedPendulumTransporterPhysics::Controllers↔  
::WindController, 66
- ResetPlots
  - InvertedPendulumTransporter::Controls::Plots↔  
Control, 46
- ResetSimulation
  - InvertedPendulumTransporter::Controls::Scene↔  
Control, 49
- ResetSystemParameters
  - InvertedPendulumTransporterPhysics::Common↔  
::SystemState, 57
- ResetTimer
  - InvertedPendulumTransporterPhysics::Common↔  
::SystemState, 57
- RodLength
  - InvertedPendulumTransporter::Models::IPendulum,  
27
  - InvertedPendulumTransporter::Windows::Main↔  
Window, 38
- SaveTrajectory
  - InvertedPendulumTransporterPhysics::Common↔  
::ITrajectoryController, 30
  - InvertedPendulumTransporterPhysics::Controllers↔  
::TrajectoryController, 61
- SceneControl
  - InvertedPendulumTransporter::Controls::Scene↔  
Control, 49
- SetAccuracy
  - InvertedPendulumTransporterPhysics::Controllers↔  
::ITrajectoryController, 30
  - InvertedPendulumTransporterPhysics::Controllers↔  
::TrajectoryController, 61
- SetAngleError
  - InvertedPendulumTransporterPhysics::Controllers↔  
::PIDCorrector, 45
- SetControlError
  - InvertedPendulumTransporterPhysics::Controllers↔  
::IVoltageController, 32
  - InvertedPendulumTransporterPhysics::Controllers↔  
::VoltageController, 64
- SetPositionError
  - InvertedPendulumTransporterPhysics::Controllers↔  
::PIDCorrector, 45
- SetTime
  - InvertedPendulumTransporterPhysics::Controllers↔  
::IVoltageController, 32
  - InvertedPendulumTransporterPhysics::Controllers↔  
::VoltageController, 64
- SetUserAngle
  - InvertedPendulumTransporterPhysics::Controllers↔  
::IVoltageController, 32
  - InvertedPendulumTransporterPhysics::Controllers↔  
::VoltageController, 65
- SetupHelpWindow
  - InvertedPendulumTransporter::Windows::About↔  
Window, 11
- SetupHighLevelGraphics
  - InvertedPendulumTransporter::Controls::Scene↔  
Control, 49
  - InvertedPendulumTransporter::Models::Cart, 13
  - InvertedPendulumTransporter::Models::IModel, 21
  - InvertedPendulumTransporter::Models::Pendulum,  
43
  - InvertedPendulumTransporter::Models::Simulation↔  
Area, 52
- SetupLowLevelGraphics
  - InvertedPendulumTransporter::Controls::Scene↔  
Control, 49
  - InvertedPendulumTransporter::Models::Cart, 13
  - InvertedPendulumTransporter::Models::IModel, 22
  - InvertedPendulumTransporter::Models::Pendulum,  
43
  - InvertedPendulumTransporter::Models::Simulation↔  
Area, 52
- SetupStrategy
  - InvertedPendulumTransporterPhysics::Solvers::I↔  
ODESolver, 23
  - InvertedPendulumTransporterPhysics::Solvers::↔



- ODESolver, [39](#)
- SetupWindowType
  - InvertedPendulumTransporter::Windows::About↔  
Window, [12](#)
- ShowCartTrajectory
  - InvertedPendulumTransporter::Controls::Scene↔  
Control, [49](#)
- ShowPendulumTrajectory
  - InvertedPendulumTransporter::Controls::Scene↔  
Control, [50](#)
- ShowTargetTrajectory
  - InvertedPendulumTransporter::Controls::Scene↔  
Control, [50](#)
- SimulationArea
  - InvertedPendulumTransporter::Models::Simulation↔  
Area, [52](#)
- simulationArea
  - InvertedPendulumTransporter::Controls::Scene↔  
Control, [51](#)
- Size
  - InvertedPendulumTransporter::Models::ISimulation↔  
Area, [27](#)
- SolveODESystem
  - InvertedPendulumTransporterPhysics::Solvers::I↔  
ODESolver, [25](#)
  - InvertedPendulumTransporterPhysics::Solvers::↔  
ODESolver, [39](#)
- SolverParameters
  - InvertedPendulumTransporterPhysics::Common↔  
::SolverParameters, [53](#)
  - InvertedPendulumTransporterPhysics::Common↔  
::SystemState, [58](#)
- StateX
  - InvertedPendulumTransporterPhysics::Common↔  
::SystemState, [58](#)
- StateY
  - InvertedPendulumTransporterPhysics::Common↔  
::SystemState, [58](#)
- SystemState
  - InvertedPendulumTransporterPhysics::Common↔  
::SystemState, [56](#)
- Time
  - InvertedPendulumTransporterPhysics::Common↔  
::SystemState, [58](#)
- TimeDelta
  - InvertedPendulumTransporter::Windows::Main↔  
Window, [38](#)
  - InvertedPendulumTransporterPhysics::Common↔  
::SystemState, [58](#)
- ToStateArray
  - InvertedPendulumTransporterPhysics::Common↔  
::OneDimensionalSystemState, [41](#)
- ToTimeArray
  - InvertedPendulumTransporterPhysics::Common↔  
::SystemState, [57](#)
- TrajectoryAchieved
  - InvertedPendulumTransporterPhysics::Controllers↔  
::ITrajectoryController, [31](#)
- TrajectoryController
  - InvertedPendulumTransporterPhysics::Controllers↔  
::TrajectoryController, [60](#)
- TrajectoryEnabled
  - InvertedPendulumTransporterPhysics::Controllers↔  
::ITrajectoryController, [31](#)
- TrajectoryLoaded
  - InvertedPendulumTransporter::Windows::Create↔  
TrajectoryWindow, [16](#)
- TrajectoryPoints
  - InvertedPendulumTransporter::Windows::Create↔  
TrajectoryWindow, [16](#)
- UpdateAngleErrorPlots
  - InvertedPendulumTransporter::Controls::Plots↔  
Control, [46](#)
- UpdateCamera
  - InvertedPendulumTransporter::Controls::Scene↔  
Control, [50](#)
- UpdateFrame
  - InvertedPendulumTransporter::Controls::Scene↔  
Control, [50](#)
- UpdatePositionErrorPlots
  - InvertedPendulumTransporter::Controls::Plots↔  
Control, [47](#)
- UpdateState
  - InvertedPendulumTransporter::Controls::Scene↔  
Control, [50](#)
  - InvertedPendulumTransporter::Models::Cart, [13](#)
  - InvertedPendulumTransporter::Models::IModel, [22](#)
  - InvertedPendulumTransporter::Models::Pendulum,  
[43](#)
  - InvertedPendulumTransporter::Models::Simulation↔  
Area, [52](#)
- UpdateSystemParameters
  - InvertedPendulumTransporterPhysics::Solvers::I↔  
ODESolver, [25](#)
  - InvertedPendulumTransporterPhysics::Solvers::↔  
ODESolver, [40](#)
- UpdateSystemStateX
  - InvertedPendulumTransporterPhysics::Common↔  
::SystemState, [57](#)
- UpdateSystemStateY
  - InvertedPendulumTransporterPhysics::Common↔  
::SystemState, [57](#)
- UpdateTimer
  - InvertedPendulumTransporterPhysics::Common↔  
::SystemState, [58](#)
- UpdateTrajectory
  - InvertedPendulumTransporter::Controls::Scene↔  
Control, [50](#)
- UpdateVoltagePlots
  - InvertedPendulumTransporter::Controls::Plots↔  
Control, [47](#)
- UpdateWindDirection
  - InvertedPendulumTransporter::Controls::Scene↔  
Control, [51](#)
- UpdateWindForce

- InvertedPendulumTransporterPhysics::Controllers↔
  - ::IWindController, [34](#)
- InvertedPendulumTransporterPhysics::Controllers↔
  - ::WindController, [67](#)
- UserAngleX
  - InvertedPendulumTransporterPhysics::Controllers↔
    - ::IGameController, [20](#)
- UserAngleY
  - InvertedPendulumTransporterPhysics::Controllers↔
    - ::IGameController, [21](#)
- Velocity
  - InvertedPendulumTransporterPhysics::Common↔
    - ::OneDimensionalSystemState, [41](#)
- VerticalWindForce
  - InvertedPendulumTransporterPhysics::Common↔
    - ::SolverParameters, [54](#)
- Voltage
  - InvertedPendulumTransporterPhysics::Common↔
    - ::SolverParameters, [54](#)
- VoltageController
  - InvertedPendulumTransporterPhysics::Controllers↔
    - ::VoltageController, [64](#)
- VoltagePointsX
  - InvertedPendulumTransporter::Controls::Plots↔
    - Control, [47](#)
- VoltagePointsY
  - InvertedPendulumTransporter::Controls::Plots↔
    - Control, [47](#)
- wheelRadius
  - InvertedPendulumTransporter::Models::Cart, [15](#)
- WindChangeSpeed
  - InvertedPendulumTransporter::Windows::Main↔
    - Window, [38](#)
  - InvertedPendulumTransporterPhysics::Controllers↔
    - ::IWindController, [35](#)
- WindController
  - InvertedPendulumTransporterPhysics::Controllers↔
    - ::WindController, [66](#)
- WindPower
  - InvertedPendulumTransporter::Windows::Main↔
    - Window, [38](#)
  - InvertedPendulumTransporterPhysics::Controllers↔
    - ::IWindController, [35](#)
- WindType
  - InvertedPendulumTransporterPhysics::Controllers, [10](#)
  - InvertedPendulumTransporterPhysics::Controllers↔
    - ::IWindController, [35](#)
- WindowType
  - InvertedPendulumTransporter::Windows, [8](#)
- XCoordAngle
  - InvertedPendulumTransporter::Windows::Main↔
    - Window, [38](#)
- YCoordAngle
  - InvertedPendulumTransporter::Windows::Main↔
    - Window, [38](#)