Федеральное государственное бюджетное образовательное учреждение высшего образования «Комсомольский-на-Амуре государственный университет»

На правах рукописи

Append

Хрульков Владимир Николаевич

## РАЗРАБОТКА И ИССЛЕДОВАНИЕ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ УПРАВЛЕНИЯ КОМБИНИРОВАННЫМ ТРАЛОВЫМ МЕХАНИЗМОМ ДЛЯ ПЕРЕМЕЩЕНИЯ ВЫСОКОТОННАЖНЫХ КРУПНОГАБАРИТНЫХ ГРУЗОВ

Специальность 2.3.3 – Автоматизация и управление технологическими процессами и производствами (технические науки)

## ДИССЕРТАЦИЯ

на соискание ученой степени кандидата технических наук

Научный руководитель: доктор технических наук, доцент Черный Сергей Петрович

Комсомольск-на-Амуре – 2025

### Оглавление

Введение
Глава 1 Аналитический обзор способов, устройств и систем транспортировки
высокотоннажных крупногабаритных грузов 11
1.1 Особенности технологического процесса транспортировки
крупногабаритных высокотоннажных моногрузов 12
1.2 Анализ интеллектуальных методов управления сложными
многокоординатными технологическими объектами 24
Выводы по первой главе
Глава 2 Моделирование элементов распределенной гидравлической системы
модульного тралового механизма 29
2.1 Математическое описание элементов гидравлической системы 29
2.2 Моделирование системы управления активным элементом подвесной
системы трала
2.3 Математическое описание многокаскадного нечеткого логического
регулятора для многоопорного транспортного средства 46
Выводы по второй главе53
Глава 3 Синтез и анализ имитационной модели гидравлической системы
платформы модульного тралового механизма с многокаскадным
нечетким логическим регулятором 54
3.1 Определение ограничений, накладываемых на систему управления
четырехсекционной транспортной платформой
3.2 Синтез алгоритма работы системы управления
Выводы по третьей главе111
Глава 4 Практическая реализация нечетких алгоритмов управления
с использованием промышленных средств автоматизации 112
Выводы по четвертой главе115
Заключение117
Список использованных источников 119

Триложение А (обязательное) Акты внедрения					
Приложение Б (обязательное) Свидетельства регистраций программ					
для ЭВМ 135					
Приложение В (обязательное) Код программы многокаскадного нечеткого					
логического регулятора, формирующий уставки					
внутренних модулей за счет суммирования					
значений сигналов управления 138					

#### Введение

Актуальность. Транспортировка крупногабаритных и сверхтяжелых грузов по дорогам общего пользования с помощью модульных траловых механизмов представляет собой существенную проблему, связанную не только с наличием целого ряда технических особенностей, но и большим числом экономических и организационно-логистических задач. К таким техническим особенностям можно отнести: кривизну дорожного покрытия, отсутствие крепления грузов к платформе, высокий центр тяжести перевозимого груза, большую парусность. К экономическим и организационно-логистическим задачам относят ограничение трафика движения, разветвленность дорожной сети и удаленность места будущего использования. В настоящее время большинство задач, связанных с перечисленными ограничениями, решаются путем поиска альтернативных вариантов доставки, вплоть до строительства специального отдельного дорожного полотна, что представляет собой решения, ориентированные на уникальные строго определенные условия. Из-за несовершенства системы управления подвеской, основанной на классических методах регулирования, которая ориентирована на реализацию процедур управления отдельным узлом подвесной системы, нередки случаи, когда транспортируемый груз подвергался опрокидыванию, что приводило к повреждению самого перевозимого объекта, наносило непоправимый ущерб модульному траловому механизму, разрушало дорожное полотно. Совокупность обозначенных особенностей объекта регулирования связана с отсутствием комплексного подхода к реализации системы управления сложным многокоординатным объектом, таким как модульный траловый механизм, при его эксплуатации по дорогам общего пользования путем контроля ряда недетерминированных параметров. Таким образом, представляется актуальным применение интеллектуальных методов управления гидравлической подвеской, которые позволят решить большую часть проблем технического характера, связанных с наличием различных возмущающих воздействий, обусловленных как природным характером, так и вариацией технологических параметров.

Научная новизна предложенного подхода по созданию интеллектуальной системы управления основана на использовании комплекса специализированных нечетких модулей, образующих единую иерархическую многокаскадную структуру. Применение многокаскадной нечеткой системы управления таким сложным технологическим объектом, как модульный траловый механизм, позволит существенно повысить возможности его применения на дорогах общего пользования и снизить возможные риски, связанные с рядом внешних и внутренних возмущений, а также обеспечить независимость системы регулирования от компоновки его составных частей.

Научная новизна предлагаемых решений заключается в следующем:

1) предложено структурное решение и математическое описание многокаскадного нечеткого логического регулятора на основе операторов произведения и суммы для вложенных каскадов, которое позволяет сформировать требуемые зависимости между основными параметрами технологического процесса по перемещению высокотоннажных крупногабаритных грузов;

2) модель многокаскадной нечеткой системы управления модульным траловым механизмом, позволяющая повысить возможности по перемещению объекта управления под влияние недетерминированных факторов;

 алгоритм функционирования системы управления процессом стабилизации модульным траловым механизмом, отличающийся от ранее известных тем, что формирование управляющих процедур осуществляется многокаскадным нечетким логическим регулятором.

#### Степень разработанности темы исследования

На сегодняшний день глубоко изучены способы поддержания заданного положения в пространстве гидроприводом, однако ряд способов имеют технологические ограничения и не обладают предиктивным управлением. Задачей управления процессом стабилизации высокотоннажных крупногабаритных грузов на передвижном модульном траловом механизме является поддержание стабильного устойчивого положения перемещаемого объекта в условиях недетерминированных возмущений, формируемых внешними факторами, что не позволяет исполь-

зовать классические методы управления локально на нижнем уровне регулирования каждым отдельным активным элементом подвесной системы.

Работа основывается на достижениях отечественных и зарубежных ученых в области проектирования гидравлических систем, а также автоматизированных и интеллектуальных систем управления технологическими процессами: Опейко А.Ф., Белоусов Б.Н., Попов С.Д., Аверьянов О.И., Ермакова С.А., Амиров Ю.Д., Башта Т.М., Бесекерский В.А., Гинцбург Л.Л., Домогаров А.Ю., Гладов Г.И., Вихров А.В., Зайцев С.В., Кувшинов В.В., Павлов В.В., Афанасьев Б.А., Бочаров Н.Ф., Жеглов Л.Ф., Зузов В.Н., Полунгян А.А., Фоминых А.Б., Цыбин В.С., Навроцкий К.Л., Попов Д.Н., Штовба С.Д., Кудинов Ю.И., Мелихов А.Н., Блишун А.Ф., Захаров В.Н., Орлов А.И., Поспелов Д.А., Троценко В.В., Hrovat D., Mahmoud Omar, El-kassaby М.М., Walid Abdelghaffar, Musa Mohammed Bello, Amir Akramin Shafie, Raisuddin Khan, M. Senthil Kumar, Vijayarangan S., Asai K., Zadeh L., Sugeno M., Takagi T., Mamdani E.A.

**Объект исследования** – система автоматического управления гидравлической подвеской модульного тралового механизма, обеспечивающая стабилизацию транспортируемого объекта.

**Предмет исследования** – процедуры управления, реализуемые многокаскадным нечетким регулятором, для формирования сложных законов управления подвесной системой модульного тралового механизма.

**Цель работы** – реализация сложных законов управления многокаскадным нечетким логическим регулятором по стабилизации комбинированного тралового механизма при перемещении высокотоннажных крупногабаритных грузов в движении с учетом действия недетерминированных возмущающих воздействий.

#### Задачи диссертации:

 анализ существующих подходов в области управления составной многокоординатной системой на примере модульного тралового механизма, учитывающего функционирование объекта в условиях случайных возмущений и нестационарности параметров; 2) разработка математического описания многокаскадного нечеткого логического регулятора, отличающегося структурой построения, особенностью которой является формирование уставок для регуляторов вложенного каскада на основе операторов произведения и суммы;

 разработка имитационной модели многокаскадной нечеткой системы управления, которая позволяет учесть продольные и поперечные крены составляющих полуплатформ и реализовать стабилизацию всех элементов модульного тралового механизма;

4) апробация предложенного алгоритма управления с использованием средств промышленной автоматики на имитационной модели.

#### Методология и методы исследования

В процессе синтеза и исследования интеллектуальной системы управления модульным траловым механизмом для перемещения высокотоннажных крупногабаритных грузов широко использовались методы математического анализа, теории автоматического управления и моделирования для описания динамики исследуемой системы, методы теории нечетких множеств при реализации сложных законов регулирования.

Область исследования соответствует паспорту научной специальности 2.3.3 Автоматизация и управление технологическими процессами и производствами (технические науки) по пунктам:

-«п. 4. Теоретические основы и методы моделирования, формализованного описания, оптимального проектирования и управления технологическими процессами и производствами»;

-«п. 5. Научные основы, алгоритмическое обеспечение и методы анализа и синтеза систем автоматизированного управления технологическими объектами»;

-«п. 6. Научные основы и методы построения интеллектуальных систем управления технологическими процессами и производствами».

#### Положения, выносимые на защиту:

1) структурное решение и математическое описание многокаскадного нечеткого логического регулятора, реализующего сложный закон управления мо-

дульным траловым механизмом, особенностью которого является синтез уставок для регуляторов вложенного каскада на основе операторов произведения и суммы, позволяющего сформировать требуемые функциональные зависимости между параметрами, оказывающими наибольшее влияние на технологический процесс;

2) модель многокаскадной нечеткой системы управления модульным траловым механизмом, позволяющая учесть влияние большинства внешних факторов и повысить возможности по передвижению механизмов в условиях дорог общего пользования;

3) алгоритм функционирования системы управления процессом стабилизации модульным траловым механизмом, отличающийся от ранее известных тем, что формирование управляющего воздействия осуществляется с помощью многокаскадного нечеткого логического регулятора на основании контроля входных параметров, позволяющих стабилизировать транспортируемый объект.

**Практическая значимость** достигается разработанной системой управления процессом стабилизации модульного тралового механизма для перемещения высокотоннажных крупногабаритных грузов, обладающей возможностью снижения негативного влияния ряда возмущающих факторов технического и технологического плана, что позволяет сократить время транспортировки, повысить надежность и сохранность технологического оборудования. Практическая значимость подтверждается внедрением полученных результатов в части алгоритмического и программного обеспечения интеллектуальных систем многоопорных транспортировочных машин и подкреплена актом внедрения на Комсомольской дистанции электроснабжения Дальневосточной дирекции по энергообеспечению ОАО «РЖД», а также в учебном процессе кафедры «Электропривод и автоматизация промышленных установок» ФГБОУ ВО «Комсомольский-на-Амуре государственный университет».

#### Апробация работы

Результаты диссертационной работы докладывались и обсуждались на IV Международной научно-практической конференции «Производственные технологии будущего: от создания к внедрению» (Комсомольск-на-Амуре, 2021); V Меж-

дународной научно-практической конференции «Производственные технологии будущего: от создания к внедрению» (Комсомольск-на-Амуре, 2022); VI Международной научно-практической конференции «Производственные технологии будущего: от создания к внедрению» (Комсомольск-на-Амуре, 2023); III Всероссийской национальной научной конференции студентов, аспирантов и молодых ученых «Молодежь и наука: актуальные проблемы фундаментальных и прикладных исследований» (Комсомольск-на-Амуре, 2020); Международной молодежной научной конференции «Гагаринские чтения – 2020» (Москва, 2020); IV Всероссийской национальной научной конференции студентов, аспирантов и молодых ученых «Молодежь и наука: актуальные проблемы фундаментальных и прикладных исследований» (Комсомольск-на-Амуре, 2021); XLVII Международной молодежной научной конференции «Гагаринские чтения – 2021» (Москва, 2021); 2020 International Multi-Conference on Industrial Engineering and Modern Technologies, FarEastCon 2020 (Владивосток, 2020); XXIV Краевом конкурсе молодых ученых и аспирантов Хабаровского края (Хабаровск, 2021); Внутривузовском конкурсе молодых ученых в сфере научных исследований (Комсомольск-на-Амуре, 2022); 2023 International Russian Automation Conference (RusAutoCon; Coчи, 2023); II Международной научно-практической конференции молодых ученых (Комсомольск-на-Амуре, 2023); научно-техническом семинаре высшей школы кибернетики и цифровых технологий ФГБОУ ВО «ТОГУ» – 2024 (Хабаровск, 2024).

Научные исследования выполнялись в рамках НИР № ВН010/2021 «Разработка принципов построения интеллектуальных систем управления сложными техническими объектами на основе критериев энергоэффективности».

**Личный вклад автора** заключается в постановке цели и задач диссертационного исследования; анализе отечественных и зарубежных научных публикаций по теме исследования; в проведении имитационного моделирования и вычислительного эксперимента с применением общепромышленных средств автоматизации; в разработке алгоритмов нечеткого многокаскадного управления стабилизацией комбинированного тралового механизма.

Публикации, патенты и свидетельства. Основные результаты исследования опубликованы в 20 работах, в том числе 4 – в изданиях из перечня, рекомендованного ВАК, 5 – в изданиях, индексируемых в международных базах SCOPUS и Web of Science, 3 свидетельства о регистрации программы для ЭВМ.

Структура и объем работы. Диссертация состоит из введения, четырех глав, заключения, списка использованных источников, состоящего из 129 наименований, и 3 приложений. Основная часть диссертационной работы изложена на 132 страницах, содержит 88 рисунков и 1 таблицу.

# Глава 1 Аналитический обзор способов, устройств и систем транспортировки высокотоннажных крупногабаритных грузов

Модульные траловые механизмы особо большой грузоподъемности имеют огромное значение для экономики, энергетики и тяжелой промышленности любой страны. С середины XX в. одним из способов повышения эффективности грузового автомобиля (многоосевого тралового механизма) является увеличение его грузоподъемности. Грузоподъемность многоопорных траловых механизмов достигалась за счет:

1) увеличения числа осей от 4-х до 12;

2) увеличения опор на траловой платформе – более 20, и количества колес более 100, обеспечивающих грузоподъемность до 500 т и более;

3) комбинирования многоопорных машин последовательно (автопоезда).

Многоопорные автомобили способны перевозить к месту назначения модули цехов, фабрик и заводов, морских доков и других сооружений больших размеров и масс. К таким транспортерам предъявляются специальные требования: обеспечение проезжаемости с многотонным негабаритным грузом по существующей дорожной сети, создание возможностей для размещения и закрепления особого груза, а также его погрузки и разгрузки [40, 41].

Перевозка составных крупногабаритных грузов и моногрузов (тяжелые неделимые объекты) по автомагистралям общего назначения представляет собой существенную проблему, связанную не только с большим перечнем технических задач, но и с множеством организационно-логистических и экономических задач. К проблемам технического характера можно отнести следующее: состояние дорожного полотна, высокую парусность транспортируемого груза из-за габаритных особенностей, отсутствие жесткого крепления груза к специализированному транспортировочному устройству, угол наклона дорожного полотна. К категории экономических и организационно-логистических задач можно отнести: отдаленное местоположение станций технического обслуживания от места использования, перекрытие автомобильного трафика на шоссе, разветвленность дорожнотранспортной сети, строительство дорожного полотна специального назначения.

# 1.1 Особенности технологического процесса транспортировки крупногабаритных высокотоннажных моногрузов

На данный момент описанные выше задачи и проблемы решаются путем загрузки уже имеющихся подъездных путей либо строительством дополнительных специализированных дорожных сетей, изготовлением специализированных транспортных модулей, а также поиском альтернативных вариантов транспортировки. Большинство транспортных компаний зачастую решают данную проблему путем разделения транспортируемого объекта на узлы, тем самым уменьшая вес и распределяя его по нескольким транспортным платформам [83].

Анализируя сферы применения модульных траловых механизмов для перемещения сверхтяжелых грузов, можно выделить промышленные отрасли их использования, в числе которых электроэнергетика (транспортировка генератора электростанции, электрические трансформаторы, детали ветрогенераторов), горнодобывающая промышленность (транспортировка карьерных самосвалов, шагающих экскаваторов, бульдозеров), нефтеперерабатывающая отрасль (транспортировка реактора гидрокрекинга, нефтяных установок, ректификационных колонн).

Например, горнодобывающая промышленность является основным средством обеспечения материалов для ювелирной промышленности, промышленности производства строительных материалов (гранит, цемент), черной и цветной металлургии, которые относятся к первичному сектору экономики. В процессе развития технологического оборудования для добычи полезных ископаемых объемы и разнообразие сырья кратно увеличились, тем самым сформировались новые требования по максимальному количеству перевозимого сырья транспортировочными машинами до пункта переработки и сортировки. Исходя из образовавшихся условий, такими компаниями как Komatsu, Liebherr, Caterpillar, Unit Rig (Terex), Белаз были спроектированы и разработаны сверхтяжелые карьерные самосвалы, грузоподъемность которых составляет более 300 т.

В связи со спецификой использования и возрастающими требованиями по транспортируемому количеству полезных ископаемых карьерные самосвалы приобрели крупные габариты и большой вес, в результате чего возникла проблема транспортировки их до места проведения работ, так как вес машины составлял уже более 600 т. Самостоятельное перемещение по дорогам общего пользования для них стало запрещено из-за деформации дорожного полотна, что делает невозможным самостоятельное перемещение грузовых машин своим ходом по транспортным магистралям. В крайних случаях машина транспортируется своим ходом по специально подготовленному полотну, но даже в этом случае вес машины без нагрузки, который составляет от 260 до 360 т, распределен на четыре колеса, наносит серьезные повреждения дорожному покрытию, что приводит к дополнительным финансовым затратам на его поддержание в надлежащем состоянии.

В таблице 1.1 представлены наиболее часто используемые грузовые машины на горно-перерабатывающем производстве.

Таблица	1.1 –	Массогабаритные	характеристики	карьерных	самосвалов
		без нагрузки			

Произволители	Молени	Габаритные размеры	Вес машины без
производитель	модель	самосвалов, мм	нагрузки, т
Liebherr	T 282 B	15 300×9520×7840	252
Komatsu	980E	13 900×10 000×8000	255
<b>БЕЛАЗ</b>	75710	20 600×9750×8170	390
Caterpillar	797F	14 802×9529×6998	260
Unit Rig (Terex)	MT 6300AC	15 570×9700×7920	235

Из проведенного анализа можно констатировать, что транспортировка сверхтяжелых крупногабаритных грузов может осуществляться следующими способами: 1 Транспортировка машин по воде с помощью специализированных барж и грузовых суден. В данном случае машина транспортируется в собранном состоянии при условии, что рядом с местом выгрузки находится карьер и машина может своим ходом до него доехать. Иначе машина разбирается на составные части и последующая сборка осуществляется на станции технического обслуживания.

2 Транспортировка с помощью железнодорожного транспорта. При этом способе перевозки самосвал делят на узлы, которые по весовым параметрам и габаритам разрешено транспортировать по железнодорожному полотну. После этого разобранную машину доставляют до ближайшей железнодорожной станции, находящейся рядом с будущим местом использования. Последующая сборка машины может осуществляться непосредственно на самой железнодорожной станции, после чего самосвал своим ходом перемещается к месту работы или разобранная машина загружается на специализированные транспортировочные устройства, которые доставляют ее на станции технического обслуживания (находящиеся в непосредственной близости от карьера), где специалисты производят пусконаладку машины.

3 Транспортировка карьерного самосвала с помощью специализированного транспортировочного устройства. Таким устройством является комбинированный траловый механизм, который делает возможным, не разбирая машину на узлы, доставить его до места работы по автомагистралям общего назначения [88].

В настоящее время такие компании, как Cometto, DTA, Mammoet являются лидерами на рынке по транспортировке крупногабаритных высокотоннажных грузов по автомагистралям общего пользования, используя подход модульной компоновки траловых механизмов для перевозки объектов без разделения их на составные части и узлы. Они решают описанную выше проблему с помощью распределения массы транспортируемого объекта в несколько сотен тонн на некотором числе жестко соединенных между собой траловых механизмов. Общая длина может достигать 32 м и иметь 22 оси (рисунок 1.1).



Рисунок 1.1 – Собранный транспортировочный модуль, состоящий из 14 полуприцепов для транспортировки сверхтяжелых грузов

По своей конструкции все модульные траловые механизмы имеют схожие принципы построения. Они представляют собой конструкцию из самоходных модулей с большим числом колес, количество которых может варьироваться от габаритных размеров и веса транспортируемого моногруза. В зависимости от конфигурации и поставленных задач по транспортировке груза на одном из модулей могут размещаться кабина и дизельный двигатель. Обобщённо модульный траловый механизм состоит из жесткой платформы (станины), которая поддерживается гидравлическими маятниковыми осями, крепящимися попарно. Каждая маятниковая ось (рисунок 1.2) может состоять из гидродвигателя, поворотного и силового гидроцилиндров, двух или четырех колес и шарнирного коленчатого соединения. Гидроцилиндр и шарнирное колено обеспечивают подъем платформы, а также служат системой подвески. Гидравлическая система также позволяет распределять нагрузку на несколько маятниковых осей в условиях неровной поверхности. Для обеспечения высокой маневренности все маятниковые опоры могут поворачиваться на 90°. Ход подвески опоры может изменяться в пределах от 300 до 1000 мм. Грузоподъемность одной опоры с четырьмя колесами достигает 25 т.



Рисунок 1.2 – Колесная опора маятникового типа

Основным узлом модульного тралового механизма является гидростатическая трансмиссия. С помощью нее обеспечиваются привод колес (в зависимости от конфигурации трала и веса транспортируемого груза), подъем и сохранение платформой горизонтального положения (горизонтирование), торможение транспортера. От двигателя мощность подводится непосредственно к валу насоса, который создает высокое (35-100 МПа) давление в нагнетающей магистрали. В связи с тем, что основная задача, которую должен выполнять модульный траловый механизм, связана с горизонтированием платформы, в гидравлической системе предусмотрена возможность отбора потока жидкости для привода системы управления (поворотные гидроцилиндры) и подъема грузовой платформы (силовые гидроцилиндры). Отработанное гидравлическое масло в двигателях и других системах по магистралям низкого давления обратно поступает в гидравлическую станцию [1]. Для компенсации утечки, а также для охлаждения масла трансмиссия оснащена системой подпитки и охлаждения, состоящей из подпиточного насоса, фильтров и охладителей рабочей жидкости.

На данный момент для перемещения крупногабаритных сверхтяжелых грузов распространены два типа модульных траловых механизма: 1) активный модуль с наличием силового агрегата и кабины водителя, в котором осуществляется управление перемещением и контроль подвесной системы;

2) пассивный – отсутствуют кабина водителя и силовой агрегат для самостоятельного перемещения, управление гидроприводами подвесной системы либо ручное, либо автоматическое [12, 38].

Рассмотрим поэтапную погрузку карьерного самосвала на пассивный модульный траловый механизм и последующую его транспортировку, так как этот объект является наиболее часто транспортируемым.

Перед сборкой прицепа рабочий производит визуальный осмотр полуприцепа, где проверяет уровень масла в гидравлической системе, состояние гидроцилиндров, гидрораспределителей и давления шин. Убедившись, что все элементы гидравлической системы находятся в исправном состоянии и готовы к эксплуатации, специалист выравнивает платформу с помощью ручного управления гидравлической системой по горизонту до необходимого уровня, готовя ее к стыковке с остальными полуплатформами.

Далее на собранную из некоторого числа тралов передвижную платформу ставят раму (на которую в дальнейшем устанавливается груз) и перевозят ее на место подготовки (рисунок 1.3). Эта рама необходима в связи с тем, что если попытаться заехать на прицеп своим ходом, то передняя часть платформы, на которой нет нагрузки, начнет подниматься вверх, тем самым приводя к поломке соединительных узлов между платформами и гидравлических цилиндров у начала тягача из-за высокой степени нагрузки.



Рисунок 1.3 – Опорные сооружения для рамы передвижного трала

Для того чтобы снять раму с прицепа по ее краям монтируются опорные сооружения из бруса, а расположенные в этих местах гидропривода, которые имеют ход поршня в 100 мм, позволяют ей подниматься. После того как рама стала выше, чем передвижная платформа, платформу убирают и опускают на землю для дальнейшей погрузки на нее груза (рисунок 1.4). Груженую раму с помощью опорных сооружений вновь поднимают на необходимую высоту и погружают ее уже на передвижной трал. Как только рама заняла свою позицию, рабочийналадчик проверяет состояние всей конструкции в целом и проводит окончательную подстройку гидравлической системы, после чего начинается транспортировка до пункта назначения.



Рисунок 1.4 – Подъем рамы с грузом

В процессе транспортировки на модульном траловом механизме находится оператор, который в режиме реального времени производит ручную корректировку положения платформы на протяжении всего этапа доставки и контролирует общее состояние гидравлической системы. Отсутствие адаптивной системы управления и длительное ручное управление оператором могут привести к нештатным случаям (например, выходу из строя какого-либо элемента гидравлической системы трала), что существенно замедлит процесс транспортировки объекта и увеличит финансовые затраты из-за задержек (рисунок 1.5).



Рисунок 1.5 – Пример нештатной ситуации при выходе из строя элемента подвесной системы

Анализируя приведенный пример, можно сделать вывод о том, что характеристики груза (размеры, вес и т.д.) и маршрут перемещения определяют количество и конфигурацию используемых модулей гидравлической системы. После определения количества осей и типа прицепа необходимо решить, как оптимально сгруппировать баллоны для обеспечения максимальной устойчивости транспортировки. Каждый модуль прицепа оснащен гидравлической системой, включающей различные клапаны, трубы и цилиндры, позволяющие гидравлически соединять несколько прицепов. Для группировки цилиндров используются пропорциональные гидравлические распределители, расположенные под платформой прицепа. Направление движения колесной пары при повороте транспортировочной платформы приведено на рисунке 1.6.



Рисунок 1.6 – Направление движения колесной пары при повороте транспортировочной платформы

Одним из вариантов транспортировки моногруза с помощью активного многоопорного транспортного средства можно рассматривать транспортировку реактора гидрокрекинга, которая проходила в ноябре 2013 г. в г. Комсомольскена-Амуре. Крупногабаритный моногруз транспортировался по специально подготовленному бетонному дорожному полотну протяженностью в 7,29 км от речного порта до места установки на НПЗ НК «Роснефть Комсомольск» (рисунок 1.7). Это было обусловлено тем, что весогабаритные характеристики реактора (вес 1360 т, длина 14 м, диаметр 5 м) и система управления транспортировочного трала не позволяли провести его по дорогам общего пользования, не деформировав дорожное полотно. Как было описано ранее, одной из экономических и организационно-логистических задач является строительство дорожного полотна специального назначения. В России средняя стоимость строительства 1 км бетонной автодороги может варьироваться от 50 до 150 млн р. зависимости от ряда факторов, таких как климатические условия регионов, сложности участка, трудозатраты, стоимость материалов и техники.



Рисунок 1.7 – Карта с протяженностью бетонного дорожного полотна от речного вокзала до НПЗ НК «Роснефть Комсомольск»

Анализируя описанные выше примеры можно констатировать, что зачастую управление гидроприводами производится с помощью выносного пульта оператора, соединенного с силовой установкой, входящей в комплект поставки и включающей в себя гидростанцию с двигателем [16]. Оператор осуществляет рулевое управление тралом, а также следит за тем, чтобы платформа не отклонялась от своего горизонтального положения. При необходимости оператор корректирует положение платформы с помощью пульта, поднимая или опуская гидроцилиндры.

На текущий момент модульные траловые механизмы оснащаются комплексными электронными системами мониторинга, выводящими на дисплей в ка-

бине оператора изображение положения центра масс автомобиля при движении и погрузке/разгрузке, а также показывающими уровень загрузки опор по отношению к полной массе транспортируемого объекта. Система позволяет оператору правильно расположить груз на платформе и следить за состоянием нагруженности опор в движении. При возникновении перегрузок более 20 % от нормы подаются звуковой и световой сигналы. Наличие такой системы при перевозке высокотоннажных крупногабаритных конструкций необходимо для исключения опрокидывания груза и аварийных поломок [1, 11, 22]. Комплексная электронная система управления является в основном мониторинговой, так как большая часть поступающей информации обрабатывается самим оператором, на основании которой он корректирует положение платформы в реальном времени. В целом такой подход является времязатратным.

Независимо от компоновки транспортных многоопорных полуплатформ они оснащаются следующим перечнем датчиков: датчик скорости движения модульной полуплатформы; датчики линейных перемещений гидропривода подвески; тензометрические датчики измерения статического веса полуплатформы; датчики углов продольного и поперечного кренов колесного транспортного средства.

Проанализировав функционирование модульного тралового механизма, можно выделить ряд возмущающих факторов, оказывающих наибольшее влияние на объект (рисунок 1.8):

- скорость перемещения платформы (V);

- углы продольного и поперечного кренов (α, β);

- положение штоков гидроприводов подвески (*L*);

- положение полуплатформ при горизонтальном отклонении относительно друг друга (*l*).



ГЦ11 – ГЦ22 – гидроприводы подвески трала Рисунок 1.8 – Схема действующих на объект возмущений

Учитывая требования к эксплуатации модульного тралового механизма, можно сформулировать следующую систему ограничений, действующих на объект управления:

- отсутствие возможности крепления груза и высокий центр тяжести;

- количество преодолеваемых перекрестков;

- величины радиусов проходимых поворотов.

При этом система управления объектом должна учитывать перечень недетерминированных возмущающих воздействий:

- качество дорожного покрытия магистралей общего пользования;

- кривизну дорожного покрытия в продольной и поперечной проекциях;

- особенности климатических условий функционирования (например, наличие бокового ветра, гололед, дождь, снег);

- неравномерность нагрузки на модули траловой системы [86].

В ходе анализа функционирования объекта и его системы управления, а также действующих на него недетерменированных возмущений возникает необходимость рассмотреть интеллектуальные методы управления, обладающие предективными свойствами.

## 1.2 Анализ интеллектуальных методов управления сложными многокоординатными технологическими объектами

Существует множество различных подходов для управления сложными технологическими процессами, основывающихся на технологии искусственного интеллекта [2, 95, 114]. При управлении технологическими процессами чаще всего применяются:

1) нейронные сети;

2) генетические алгоритмы;

3) нечеткая логика.

Главной особенностью настройки нейронной сети является возможность выбора различных типов архитектуры, таких как многослойные перцептроны, рекуррентные нейронные сети, сверточные нейронные сети, которые будут направлены на решение конкретной задачи в АСУ ТП. Увеличение количества скрытых слоев и нейронов в этих слоях позволит существенно повысить вычислительные возможности сети, что приведет к росту качества обработки более сложных моделей. Однако необходимо отметить, что увеличение внутренней структуры кратно усложняет алгоритм настройки, снижает быстродействие и может привести к избыточности системы управления технологическим процессом. Существует множество различных функций активации слоя, такие как сигмоидальная функция, гиперболический тангенс, Rectified Linear Unit и др. Выбор функции активации зависит от задачи, типа данных и архитектуры сети и может потребовать экспериментов для выбора оптимальной функции для конкретного случая [63, 75]. Нейронная сеть позволяет эффективно моделировать нелинейные зависимости между различными переменными в системе, что делает их полезными при работе с неструктурированными данными.

Вместе с тем нейронные сети обладают рядом недостатков, зачастую не позволяющих применять их из-за необходимости большого количества данных для обучения, что в случае ограниченности данных в АСУ ТП вызывает значительные трудности. Сложность интерпретации результатов также является недостатком применения нейронных сетей в АСУ ТП. Так как нейронные сети представляют собой модели типа «черный ящик», то зачастую представляется затруднительным определить процесс получения ей итогового результата. Преодолеть проблему извлечения знаний из самообучающихся нейронных сетей до сих пор представляется затруднительным. Также нейронные сети могут быть чувствительны к изменениям в данных, что может привести к нестабильности работы системы. Специалист в области АСУ ТП должен обладать обширными знаниями, обеспечивающими его возможности оперирования сложным математическим аппаратом, заложенным в алгоритмы настройки нейросетевых проектов [33, 43, 105].

Альтернативным направлением в области интеллектуального управления, отличающимся от нейронных сетей, является использование генетических алгоритмов. Главным их преимуществом является возможность параллельного решения нескольких задач, высокая эффективность в тех случаях, когда обучающая выборка системы представляет большой массив данных, возможность оптимизации и экстраполирования разрывных функций, а также позволяет получить ряд альтернативных решений [67].

Недостатками применения генетических алгоритмов в АСУ ТП, в частности в локальных системах управления, являются: высокая размерность обучающей выборки, которая избыточна; многократность вычислений приводит к повышенным требованиям к аппаратной части средств автоматизации; генетический алгоритм является стохастическим (случайным), а это значит, что нет никаких гарантий получить оптимальное или качественное решение. В системах управления транспортными мобильными комплексами, где решения должны приниматься в реальном времени с минимальной задержкой, это может быть неприемлемо. Также одним из ключевых недостатков применения генетических алгоритмов является сложность адаптации к динамическим изменениям. В системах управления транспортных мобильных комплексов, где окружающая среда и условия работы могут изменяться мгновенно (например, появление препятствий, изменение дорожных условий), генетические алгоритмы могут не успеть адаптироваться. Для работы в динамических системах требуется постоянное переобучение, что увеличивает вычислительную нагрузку и снижает эффективность.

Системы с нечеткой логикой позволяют успешно преодолевать проблемы, свойственные перечисленным выше технологиям интеллектуального управления, поскольку применение нечетких логических регуляторов с различными формами функций принадлежности и алгоритмами вывода обеспечивают заданное быстродействие при динамическом изменении внешних параметров. В зависимости от настройки нечеткие логические регуляторы имеют низкую вычислительную нагрузку на аппаратную платформу контроллера, позволяют настроить регуляторы при неполноте информации о функционировании объекта управления [25, 28, 128]. Нечеткие логические регуляторы в отличие от классических принципов построения, которые настраиваются в зависимости от внутренних параметров объекта и его полного или частичного математического описания, позволяют получать необходимые переходные процессы независимо от недетерминированности внешних и нестационарности внутренних возмущений. Применение НЛР в многокоординатных системах управления позволяет снизить стоимость и увеличить скорость разработки [44, 102, 118].

На основе перечня обозначенных возмущающих факторов, а также учитывая уникальность рассматриваемого объекта управления, применение классических подходов к построению системы управления представляется невозможным. Поэтому предлагается реализация законов управления сложным технологическим объектом на основе многокаскадной системы регулирования, построенной по иерархическому принципу, функциональная схема которой показана на рисунке 1.9.



Рисунок 1.9 – Функциональная схема многокаскадного нечеткого логического регулятора (НЛР)

Нижний уровень интеллектуальной системы управления представляет собой набор простейших нечетких логических регуляторов с единственными лингвистическими переменными на входе и выходе, которые реализуют функции управления отдельным элементом подвесной гидравлической системы с компенсацией возмущений по неровностям дорожного покрытия. Промежуточный уровень системы выполнен с использованием четырех нечетких регуляторов, реализующие функции стабилизации каждого отдельного модуля траловой системы в продольной и поперечной проекциях с компенсацией кривизны дорожного покрытия и неравномерности нагрузки. На верхний уровень системы возлагается главная функция, реализующая выравнивание модулей всей траловой системы относительно друг друга, что позволяет компенсировать наличие бокового ветра, а также присутствие подъемов или спусков на дорогах общего пользования [86].

#### Выводы по первой главе

Выполнен аналитический обзор существующих способов транспортировки крупногабаритных сверхтяжелых моногрузов, в том числе обзор патентнотехнической литературы. Анализ показал, что существующие способы управления комбинированным траловым механизмом отличаются отсутствием предективных свойств автоматизированной корректировки положения платформы в пространстве.

Учитывая результаты и выводы современных научных исследований в области управления сложными могокоординатными технологическими процессами, было выявлено, что актуальным является применение интеллектуальных методов регулирования. Процесс стабилизации комбинированного тралового механизма для перемещения высокотоннажных крупногабаритных грузов автоматизированной системой актуально выполнять на базе нечеткой системы, формирующей процедуры регулирования, устойчивые к влиянию внешних возмущающих воздействий, и использующей обратную связь по положению с целью минимизации колебаний объекта в движении.

Установлено, что реализацию нечеткой системы управления целесообразно проводить с использованием положений теории нечетких множеств, обладающей наиболее развитым понятийным математическим аппаратом при построении локальных систем управления.

# Глава 2 Моделирование элементов распределенной гидравлической системы модульного тралового механизма

#### 2.1 Математическое описание элементов гидравлической системы

Основным фактором применения гидропривода в многоопорных транспортных машинах является получение больших усилий и моментов, так как он обладает высокой энергонапряженностью рабочей среды при плавном равномерном движении рабочего органа и возможности бесступенчатого регулирования скорости и положения в широком диапазоне. В настоящее время для управления перемещением рабочих органов с помощью гидроприводов используют гидрораспределители (ГР) с дискретными (включено-выключено) электромагнитами и дросселирующие гидрораспределители (ДГР). Гидрораспределители выполняют по существу функции движения, реверса и остановки движения гидродвигателей [50].

Гидропривод – это довольно распространенные механизмы, которые используются во всех видах спецтехники (экскаваторы, автокраны, бульдозеры, автовышки, краны, манипуляторы, бетононасосы, погрузчики, компрессоры, самосвалы, гидромолоты, грейдеры и др.). Гидроцилиндр по принципу действия похож на пневмоцилиндр, только вместо воздуха в гидроцилиндрах движущей силой становится жидкость. Гидроцилиндры бывают двух типов действия – односторонние и двухсторонние. Односторонние цилиндры могут работать только в одном направлении. В движение они приводятся с помощью возрастающего давления рабочей жидкости в полости цилиндра. В начальное же положение они возвращаются с помощью работы пружины. Что касается двухсторонних поршней, то они мощнее односторонних, так как при приведении гидроцилиндра в действие ему не нужно преодолевать возвратную силу пружины. Кроме того, двухсторонние гидроцилиндры могут работать в двух направлениях [4, 17, 36].

Гидроприводы, в которых находят применение гидрораспределители дискретными (включено-выключено) электромагнитами, отличаются предельной простотой конструкции и, соответственно, низкой стоимостью, способны работать при повышенной степени загрязненности рабочей жидкости, однако для регулирования скорости необходима дополнительная установка дросселей или регуляторов расхода. Гидроприводы с дросселирующими гидрораспределителями отличаются превосходными точностными и динамическими характеристиками, однако они значительно сложнее, требуют сверхпрецизионного изготовления основных деталей и наивысшего качества очистки рабочей жидкости, что многократно увеличивает их стоимость и снижает надежность при эксплуатации [18-20].

В современных системах гидропривода техника пропорционального управления используется также для мягкого регулирования разгона и торможения гидродвигателей, а также для замены группы электрически переключаемых регуляторов расхода или давления одним пропорциональным аппаратом, что повышает эффективность и удобство управления системой.

Простейшим вариантом является установка пропорционального электромагнита на обычный ГР. В результате получается дополнительная возможность «приоткрывания» золотника, т.е. регулирования скорости движения гидродвигателя, однако практика показывает, что это регулирование – невысокого качества. Причины – недостаточная точность позиционирования, нестабильность расходной характеристики из-за неточности осевого расположения рабочих кромок золотника, повышенный гистерезис в результате сухого трения в подвижных элементах, зависимость расхода от перепада давлений, ограниченные динамические возможности аппарата. Для повышения качества управления проектировщиками проводились работы, связанные с усовершенствованием пропорциональных электромагнитов и управляющей электроники, установки дополнительных датчиков обратной связи (ДОС), повышения точности изготовления золотниковой пары, использования ДГР в первом каскаде усиления, введения компенсаторов перепада давлений на рабочих кромках. При этом пределом целесообразности соответствующих усложнений конструкции пропорциональных электромагнитов является сохранение конкурентоспособности по сравнению с ДГР [26, 34, 123].

Пропорциональное электроуправление находит свое основное применение в дистанционном управлении параметрами гидропривода, где мгновенное реагирование и точное регулирование играют решающую роль. Но с развитием технологий эта техника также успешно интегрируется в системы автоматического регулирования, обеспечивая совместимость с микроэлектроникой и комплексными системами управления. Современные гидрокомпоненты, снабженные встроенной электроникой, и специализированные коммутационные средства с открытой структурой позволяют объединить мощные и динамичные возможности гидравлики с быстроразвивающимися функциональностями микроэлектроники.

Достаточно хорошие перспективы применения имеются y ШИМусилителей, в которых изменение выходного сигнала достигается за счет широтно-импульсной модуляции. В результате амплитуда и частота генерируемых усилителем импульсов остаются постоянными, а ширина импульса определяется величиной входного сигнала управления. При этом пульсирующий характер сигнала, поступающего в обмотку пропорционального электромагнита, вызывает соответствующие колебания якоря и, возможно, золотника гидрораспределителя, что позволяет существенно снизить трение и улучшить характеристики аппарата в целом. Однако пульсирующий характер сигнала приводит к снижению ресурса изделия.

В распределителях с одним пропорциональным электромагнитом могут возникнуть опасные ситуации при отказе системы управления. Поэтому для повышения безопасности в ряде случаев применяют аппараты, в которых пружина в случае отказа устанавливает золотник в крайнюю позицию, обеспечивающую блокировку всех гидролиний [116, 117, 121].

Большое влияние развитие техники пропорционального управления оказывает и на современные регулируемые насосы. В частности, одной из новинок фирмы Parker является регулируемый насос с электрогидравлическим p/Qуправлением, в котором регулирование рабочего объема реализовано с помощью встроенного пропорционального распределителя с датчиком обратной связи по положению наклонного диска. В качестве компенсатора давления используется

устанавливаемый отдельно предохранительный клапан с пропорциональным электроуправлением (рисунок 2.1). Система управления имеет высокие надежность и быстродействие, она позволяет реализовать управление по давлению, а также электронное ограничение потребляемой мощности [50, 64, 73, 74, 101].



Рисунок 2.1 – Структурная схема гидропривода пропорциональным управлением

В модуле активной подвески объекта регулирования можно выделить следующие основные элементы, такие как пропорциональный гидравлический распределитель и гидравлический привод. При составлении линейной модели были приняты следующие допущения: в цилиндре и распределителе отсутствуют утечки жидкости; эффективная площадь поршня постоянна; гидравлическая жидкость несжимаема; нагрузка не обладает упругостью; расход жидкости пропорционального распределителя линейно зависит от управляющего сигнала [21, 37, 77, 78].

Гидравлический цилиндр двухстороннего действия преобразует давление жидкости в механическое движение поршня [10, 47, 115]. Его динамика может быть описана следующим образом:

$$F = A \cdot P = m \frac{d^2 x}{dt^2} + b \frac{dx}{dt} + k \cdot x, \qquad (2.1)$$

где *F* – сила, создаваемая давлением жидкости, H;

- *А* эффективная площадь поршня, м<sup>2</sup>;
- Р разность давлений в полостях цилиндра, Па;
- *m* масса поршня и нагрузки, кг;
- b коэффициент вязкого трения,  $H \cdot c/M$ ;
- *k* коэффициент упругости нагрузки, Н/м;

*х* – положение поршня, м.

Расход жидкости в гидроцилиндре двухстороннего действия *Q* связан со скоростью движения поршня

$$Q = A \cdot \frac{dx}{dt}.$$
 (2.2)

Учитывая сжимаемость жидкости, изменение давления P связано с расходом Q и объемом жидкости V в цилиндре

$$Q = \frac{V}{\beta} \cdot \frac{dP}{dt} + A \cdot \frac{dx}{dt},$$
(2.3)

где V – объем жидкости в цилиндре, м<sup>3</sup>;

β – модуль упругости жидкости, Па.

Пренебрегая упругостью нагрузки (k = 0) и сжимаемостью жидкости, передаточная функция цилиндра  $W_{\text{цил}}(p)$  между давлением P и перемещением x будет иметь следующий вид:

$$W_{\text{цил}}(p) = \frac{X(p)}{P(p)} = \frac{A}{m \cdot p^2 + b \cdot p}.$$
(2.4)

Динамика пропорционального гидравлического распределителя, регулирующего поток жидкости в цилиндр, описывается следующим образом.

Расход жидкости *Q* пропорционален управляющему сигналу *и* 

$$Q = K_v \cdot u, \tag{2.5}$$

где  $K_v$  – коэффициент пропорциональности,  $\frac{M^3/c}{B}$ ;

и – управляющий сигнал, В.

Пропорциональный распределитель имеет инерционность, связанную с электромагнитной системой управления. Его динамика может быть аппроксимирована передаточной функцией первого порядка

$$W_{\text{pacnp}}(p) = \frac{Q(p)}{U(p)} = \frac{K_{\nu}}{T(p+1)}.$$
(2.6)

Рассматривая систему «пропорциональный распределитель – гидроцилиндр» как единое целое, общая передаточная функция  $W_{\rm oбщ}(p)$  будет произведением передаточных функций распределителя и цилиндра

$$W_{\text{общ}}(p) = W_{\text{распр}}(p) \cdot W_{\text{цил}}(p) = \frac{K_{\nu}}{T \cdot p + 1} \cdot \frac{A}{m \cdot p^2 + b \cdot p}.$$
(2.7)

Введем в систему «пропорциональный распределитель – гидроцилиндр» ПИД-регулятор [6, 13, 15, 23, 35]:

$$W_{\Pi И Д}(p) = K_p + \frac{K_i}{p} + K_d \cdot p.$$
 (2.8)

Обобщенная передаточная функция системы «ПИД-регулятор – пропорциональный распределитель – гидроцилиндр» имеет вид

$$W_{\text{общ}}(p) = W_{\Pi U \square}(p) \cdot W_{\text{распр}}(p) \cdot W_{\text{цил}}(p) = \frac{A \cdot K_{\nu} \cdot (K_d \cdot p^2 + K_p \cdot p + K_i)}{p \cdot (T \cdot p + 1) \cdot (m \cdot p^2 + b \cdot p)}.$$
(2.9)

Колесные опоры комбинированных траловых механизмов являются его основным элементом, так как от их количества напрямую зависит грузоподъемность [42, 52, 100].

На рисунке 2.2 представлена принципиальная схема подвесной активной системы многоопорной машины.



Рисунок 2.2 – Принципиальная схема подвесной активной системы многоопорной машины

Колесная опора, показанная на рисунке 2.2, представляет собой фасонный шарнирный рычаг, соединенный с осью колес. В верхней части рычага установлена поворотная цапфа, соединенная с грузовой платформой. Шарниром рычаг соединяется с картером оси колес. Вторая связь верхней части рычага и картера колес осуществляется через гидравлический привод, который является упругим элементом подвески и элементом системы регулирования погрузочной высоты модульного тралового механизма [54, 61, 62, 91, 103].

Из описания конструкции опоры и элементов составных частей активной подвески модульного тралового механизма рассмотрим упрощенную кинематическую схему (рисунок 2.3).



Рисунок 2.3 – Упрощенная кинематическая схема подвески модульного трала маятникового типа

Переменные состояния подвески модульного трала: *a* – начальный угол между рамой подвески и балансиром подвески; *c* – угол поворота балансира подвески относительно рамы подвески; *x* – ход соответствующего цилиндра подвески. Исходя из кинематической схемы рисунка 2.3, можно получить следующее уравнение:

$$L_{AC} + x = \sqrt{L_{AB}^2 + L_{BC}^2 - 2L_{AB}L_{BC}\cos(a+c)}.$$
 (2.10)

В соответствии с уравнением баланса моментов подвесного механизма уравнение получается следующим образом:

$$FL_1 = GL_2, \tag{2.11}$$

где *F* – усилие, прилагаемое к подвесному цилиндру, H;

*G* – нагрузка на каждую подвеску, Н.
В соответствии с уравнением площади треугольника можно получить уравнение

$$\frac{1}{2}L_{AB}L_{BC}\sin(a+c-b) = \frac{1}{2}(L_{AC}+x)L_1.$$
(2.12)

Таким образом, *L*1 и *L*2 могут быть получены следующим образом:

$$L_{1} = \frac{L_{AB}L_{BC}\sin(a+c-b)}{\sqrt{L_{AB}^{2} + L_{BC}^{2} - 2L_{AB}L_{BC}\cos(a+c-b)}},$$
(2.13)

$$L_2 = L_{BD} \cos(a + c - j).$$
 (2.14)

Уравнение (2.11) может быть преобразовано и представлять собой соотношение между усилием подвески и усилием на одном подвесном цилиндре

$$F \cdot \frac{L_{AB}L_{BC}\sin(a+c-b)}{\sqrt{L_{AB}^2 + L_{BC}^2 - 2L_{AB}L_{BC}\cos(a+c-b)}} = G \cdot L_{BD}\cos(a+c-j).$$
(2.15)

Модульный траловый механизм представляет собой сложную многокорпусную систему, в которой существуют не только сила взаимодействия и относительное смещение компонентов, но и взаимодействие между шиной и дорогой. Поэтому очень сложно создать точную динамическую модель всего комбинированного тралового механизма в целом. В соответствии со структурой исследуемого тралового механизма при построении модели движения были приняты следующие допущения: сжимаемость жидкости можно рассматривать как пружину, а дроссельный эффект клапана и утечку из гидроцилиндра можно рассматривать как демпфирование. Упругость рамы и несущей платформы намного меньше, чем у системы подвески, и они рассматриваются как жесткие тела без учета упругой деформации. Поскольку рассматривается только система подвески транспортного средства, влияние системы рулевого управления не учитывается. Предполагается, что центр тяжести груза остается неизменным при полной загрузке [48, 65, 76, 85].

Упрощенная динамическая модель транспортного средства смоделирована так, как показано на рисунке 2.4, где *M* – это масса несущей платформы транспортного средства; *m<sub>i</sub>* – масса колеса; *k<sub>i</sub>* – жесткость гидравлической подвески;  $k_{ii}$  – жесткость шины;  $\zeta_i$  – демпфирование гидравлической подвески;  $Z_{i0}$  – вертикальное смещение точки соединения;  $z_i$  – вертикальное смещение центра шины;  $y_i$  – неровность дороги;  $Z_1$  – вертикальное смещение центра масс;  $\theta_x$  и  $\theta_y$  – угол поворота вокруг осей X и Y;  $L_1$  – ширина транспортного модуля;  $L_2$  – длина траспортного модуля.



Рисунок 2.4 – Динамическая модель модульного транспортного механизма

Согласно закону Ньютона и динамической модели транспортного средства, сила и вертикальное смещение точки соединения могут быть получены следующим образом [122]:

$$F_{i} = k_{i} \left[ z_{i} - z_{i0} \right] + ci \left[ \frac{dz_{i}}{dt} - \frac{z_{i0}}{dt} \right] (i = 1...4),$$

$$\begin{cases} z_{10} = Z_{1} - 0.5L_{1}\theta_{x} - 0.5L_{2}\theta_{y} \\ z_{20} = Z_{1} + 0.5L_{1}\theta_{x} - 0.5L_{2}\theta_{y} \\ z_{30} = Z_{1} - 0.5L_{1}\theta_{x} + 0.5L_{2}\theta_{y} \\ z_{40} = Z_{1} + 0.5L_{1}\theta_{x} + 0.5L_{2}\theta_{y} \end{cases}$$

$$(2.16)$$

Согласно уравнениям (2.16) и (2.17), три уравнения движения несущей платформы транспортного средства в трех направлениях могут быть получены в виде системы уравнений (2.18). Уравнение движения неподрессоренной массы

получается в виде уравнения (2.19).

$$\begin{cases} M \frac{d^2 Z_1}{dt^2} = \sum F_i \\ I_X \frac{d^2 \theta_X}{dt^2} = 0.5 [(F_2 + F_4) - (F_1 + F_3)] L_1, \\ I_Y \frac{d^2 \theta_Y}{dt^2} = 0.5 [(F_3 + F_4) - (F_1 + F_2)] L_2 \end{cases}$$
(2.18)

$$m_i \frac{d^2 z_i}{dt^2} + \zeta_i (z_i - z_{i0}) + k_i (z_i - z_{i0}) + k_{ti} (z_i - y_i) = 0 \ (i = 1...4),$$
(2.19)

где  $I_X$  – инерция вращения платформы вокруг оси *X*;

 $I_{Y}$  – инерция вращения платформы вокруг оси *Y*.

## 2.2 Моделирование системы управления активным элементом подвесной системы трала

На основе полученного математического описания гидравлической системы активной подвески трала осуществим моделирование активного элемента с классическим ПИД-регулированием [56, 66, 104]. На первом этапе будет осуществляться моделирование работы только одного цилиндра этой системы.

Для имитационного моделирования системы управления гидроприводами был использован пакет прикладных программ для решения задач инженерных вычислений. На рисунке 2.5 представлена структурная схема гидростанции с использованием электропривода постоянного тока, настроенного на модульный оптимум [58, 69].



Рисунок 2.5 – Структурная схема гидростанции с электроприводом постоянного тока, настроенного на модульный оптимум

Система работает следующим образом (рисунок 2.6): на электродвигатель подается сигнал управления, который начинает вращать насос и нагнетать гидравлическую жидкость в каналах из бака. Параллельно к насосу подключен напорный клапан, который контролирует давление в системе. К входу гидрораспределителя подключен сигнал синусоидальной формы, который управляет положением золотника для контроля проходимой через него жидкости. На шток привода подана нагрузка в 10 т.



Рисунок 2.6 – Система управления гидравлическим приводом

Из графика, представленного на рисунке 2.7, видно, что при постепенном перемещении золотника гидравлического распределителя шток гидравлического привода выдвигается до своего крайнего положения, равного 0,5 м, выдерживая нагрузку в 10 т. Перемещение золотника в правое крайнее положение приводит положение штока в исходное состояние [81, 97].



Рисунок 2.7 – График переходного процесса изменения положения штока

График на рисунке 2.8 демонстрирует изменение давления рабочей жидкости внутри цилиндра. Из графика видно, что в течение 10 с давление в гидроцилиндре достигает 31 МПа.



Рисунок 2.8 – График изменения давления в гидроцилиндре

Рассмотрим классическую систему управления положением гидравлического привода, реализующую ПИД-закон управления [3, 8, 14]. Основным активным элементом, реализующим такой закон управления, является пропорциональный гидравлический распределитель, позволяющий плавно регулировать поток жидкости [5, 27]. Особенностью функционирования такой системы является наличие обратной связи по положению, поступающей с датчика линейного перемещения, встроенного в привод. Информация о положении штока, поступающая на ПИДрегулятор, преобразуется в выходной сигнал, соответствующий положению золотника пропорционального гидравлического распределителя. Сигнал уставки для регулятора составляет 0,25 м, соответствующий среднему положению штока гидропривода [96-99]. Структурная схема, реализующая ПИД-закон управления, представлена на рисунке 2.9. График переходного процесса по положению гидропривода с использованием ПИД-регулятора представлен на рисунке 2.10.



1 – гидроцилиндр с датчиком линейного перемещения; 2 – пропорциональный гидравлический распределитель с ПИД-регулятором; 3 – гидравлическая станция

# Рисунок 2.9 – Имитационная модель элемента подвески модульного тралового механизма с классическим ПИД-законом управления



Рисунок 2.10 – График переходного процесса по положению гидравлического привода с ПИД-регулятором без учета возмущающего воздействия

Анализируя график переходного процесса по положению гидравлического привода с ПИД-законом регулирования без учета возмущающего воздействия, представленного на рисунке 2.10, можно сделать вывод о том, что система выходит на номинальный режим работы при реализации классического закона управления на ступенчатое входное воздействие стандартной формы. Вид и форма такого переходного процесса в системе управления гидропривода с ПИДрегулятором является эталонной для большинства активных подвесных систем [30, 96].

Рассмотрим реакцию системы управления гидравлического привода с ПИДрегулятором при подаче на него возмущающего воздействия в виде усилия в различные отрезки времени (рисунки 2.11 – 2.12).



Рисунок 2.11 – График переходного процесса по положению гидравлического привода с ПИД-регулятором при подаче на него отрицательного ступенчатого возмущающего воздействия



Рисунок 2.12 – График переходного процесса по положению гидравлического привода с ПИД-регулятором при подаче на него положительного ступенчатого возмущающего воздействия

График, представленный на рисунке 2.13, иллюстрирует реакцию системы гидропривода на возмущающее воздействие в виде усилия (рисунок 2.14), подаваемого на шток гидравлического привода. На графике видно, что при подаче возмущающего воздействия в виде усилия до 50 кН в момент времени 6,5 с шток привода выдвигается и классический ПИД-регулятор постепенно начинает компенсировать возникшее вытягивающее усилие путем изменения положения золотника гидравлического распределителя, тем самым меняя давление в полостях гидравлического цилиндра. При возникновении в момент времени 9,5 с подаваемого усилия регулятор изменяет положение золотника гидравлического распределителя и гидравлический цилиндр выходит в положение, соответствующее управляющему сигналу.



Рисунок 2.13 – График переходного процесса положения штока гидропривода с ПИД-регулятором, отрабатывающим возмущение трапецеидальной формы



Рисунок 2.14 – Возмущающее воздействие трапецеидальной формы, подаваемое на шток гидропривода

Из приведенных выше графиков можно утверждать, что модель системы работает адекватно. При построении системы будет осуществляться настройка нечеткого логического регулятора с алгоритмом вывода Такаги-Сугено нулевого порядка, который будет регулировать положение золотника гидрораспределителя для поддержания положения штока цилиндра в среднем положении.

### 2.3 Математическое описание многокаскадного нечеткого логического регулятора для многоопорного транспортного средства

Современные системы автоматического управления представляют собой сложные комплексы взаимодействующих технических устройств и элементов, работающих на различных физических принципах. Конструктивное выполнение и технические характеристики также различны.

Несмотря на многообразие отдельных систем автоматического управления и входящих в них элементов, последние могут быть сведены к нескольким основным типам, различающимся по их назначению и взаимодействию в системе управления [2, 49, 119, 120].

Мягкие вычисления – понятие, введенное Лотфи Заде в 1994 г., объединяющее в общий класс неточные, приближенные методы решения задач, зачастую не имеющие решение за полиномиальное время. Задачи, решаемые такого класса методами, возникают в области биологии, медицины, гуманитарных наук, робастного управления, менеджменте. Технологии мягких вычислений ориентированы на решение задач управления со слабо структурированными объектами управления [7, 29, 45, 55]. Инструментарий мягких вычислений использует технику нечетких систем (нечеткие множества, нечеткую логику, нечеткие регуляторы).

Целесообразно воспользоваться методами, которые специально ориентированы на построение моделей, учитывающих неполноту и неточность исходных данных. Именно в таких ситуациях технология нечеткого управления оказывается наиболее конструктивной, поскольку за последнее десятилетие на ее основе были решены сотни практических задач управления и принятия решений [60, 129].

Повышение эффективности управления является актуальной проблемой в условиях возрастающей сложности технологического оборудования, процессов и систем. Для проектирования систем управления сложными объектами важную роль играет решение задач построения адекватных математических или имитационных моделей и синтеза алгоритмов управления, обеспечивающих решение задач в условиях неопределенности. Интеллектуальные системы принятия решений (ИСПР), имитирующие принципы мышления человека, находят все более широкое применение при управлении и формализации сложных объектов и систем.

Важной отличительной чертой применения технологии мягких вычислений является ее направленность на повышение сложности процессов управления и математических моделей реальных систем, что позволяет решать задачи интеллектуального управления, подстраиваясь под достаточно существенные изменения условий внешней среды, и принимать решения по управлению, близкие к действиям эксперта или технолога.

Основное преимущество регулятора на базе нечеткой логики – это простота и наглядность формирования правил управления объектом [39, 107]. Регуляторы, построенные на базе нечеткой логики, в ряде случаев способны обеспечить более высокие показатели качества переходных процессов по сравнению с классическими регуляторами. Кроме того, используя методы синтеза нечетких алгоритмов управления, можно выполнить оптимизацию сложных контуров регулирования без проведения всесторонних математических исследований [59, 94].

Нечеткие логические регуляторы представляют собой интеллектуальные системы, имеющие в своей основе подходы, базирующиеся на экспертных оценках, именно поэтому алгоритмы синтеза таких систем трудно формализовать.

Построение регулятора, использующего многокаскадную схему нечеткого вывода, является одним из направлений развития нечетких систем управления. Реализация принципов многокаскадности позволяет повысить интеллектуальную составляющую системы, что положительно скажется на расширении диапазона применения нечетких систем управления [57, 124-127].

Рассматривая один из возможных вариантов возмущений, можно составить укрупненную структурную схему, реализующую функцию управления пропорциональными гидравлическими распределителями (рисунок 2.15) одной модульной траловой платформы. При этом интеллектуальная система регулирования должна обеспечивать стабилизацию платформы путем поддержания давления в системе пропорциональных гидравлических распределителй с учетом представленного набора детерминированных возмущений.



Рисунок 2.15 – Укрупненная структура управления положением одной модульной траловой платформы

Наличие системы управления каждым элементом подвески на локальном уровне не позволяет решить задачу управления сложным многокоординатным объектом комплексно с учетом перечисленных ранее его особенностей, в соответствии с которыми траловый механизм может состоять из нескольких полуплатформ, последовательно соединенных между собой. Каждая полуплатформа может содержать различное количество активных элементов гидравлической подвески с расположенными на них датчиками линейного перемещения, которые позволяют точно определить их положение в пространстве. Также на каждой платформе располагаются датчики горизонтального положения и датчик линейного рассогласования положения между платформами. На основе информации о конструктивных особенностях активной гидравлической подвески и параметров, поступающих на систему, была составлена функциональная схема компоновки модулей многокаскадного нечеткого логического регулятора, представленного на рисунке 2.16.



внешний каскад нечеткого регулятора; 2 – вложенный каскад первого уровня;
 3 – вложенный каскад второго уровня

### Рисунок 2.16 – Функциональная схема компоновки модулей многокаскадного нечеткого логического регулятора

Регуляторы многокаскадной нечеткой системы (см. рисунок 2.16) решают следующие задачи: 1 – внешний каскад нечеткого регулятора отслеживает рассогласование между платформами и положение крайних элементов активной подвески; 2 – вложенный каскад первого уровня отслеживает положение по оси абсцисс и ординат для каждой платформы; 3 – вложенный каскад второго уровня, управляющий положением гидроприводов активной подвески [84].

Применение нечеткой логики в гидравлической системе может позволить снизить возможные риски, связанные с неровностью дорожного покрытия, скоростью бокового ветра и выходом из строя элементов подвески [88].

50

Функционирование системы, показанной на рисунке 2.17, можно описать следующим образом. На внешний каскад нечеткого регулятора поступают сигналы с датчиков линейного рассогласования между платформами и положений крайних гидроприводов, находящихся на разных концах общей составной платформы [79]. Синтезированный корректирующий выходной сигнал с внешнего каскада регулятора формируется индивидуально для каждой полуплатформы и суммируется с сигналом управления, сгенерированным регулятором первого уровня вложенности, на который поступает сигнал положения платформы в пространстве. Суммированный сигнал внешнего регулятора и регулятора первого уровня вложенности поступает на регуляторы второго уровня вложенности, которые контролируют положение гидроприводов на транспортировочных платформах.

Функцией нечеткой системы управления с алгоритмом вывода Такаги-Сугено нулевого порядка является стабилизация положения платформы с учетом различного рода возмущающих факторов, массогабаритных параметров груза, поперечного и продольного кренов. На основе полученной информации внешний регулятор и регулятор первого уровня вложенности синтезирует сигнал управления, который изменяет положение золотников пропорциональных гидравлических распределителей, тем самым изменяя положение штоков гидроприводов.

В общем виде степени истинности для предпосылок каждого правила нечеткого регулятора внешнего каскада можно формализовать следующим образом [24, 31, 32, 51, 53, 112, 113]:

$$A_i(\Delta_{\phi}^{12}), B_i(\Delta_{\phi}^{23}), C_i(\Delta_{\phi}^{34}), D_i(L^{44}), E_i(L^{11}),$$
 (2.16)

где  $\Delta_{\phi}^{12}$ ,  $\Delta_{\phi}^{23}$ ,  $\Delta_{\phi}^{34}$  – величины отклонений между платформами;

 $L^{44}, L^{11}$  – сигналы с датчиков положения крайних гидроприводов;

*А*, *В*, *С*, *D*, *E* – входные функции принадлежности треугольной формы;

i – номер правила в базе знаний ( $i = 1...N_1$ ).

На основе операции минимума определим уровни отсечения каждой нечет-кой переменной для каждого нечеткого правила

51

$$\delta_{i} = \Lambda_{i}(A_{i}(\Delta_{\phi}^{12}); B_{i}(\Delta_{\phi}^{23}); C_{i}(\Delta_{\phi}^{34}); D_{i}(L^{44}); E_{i}(L^{11})).$$
(2.17)

Тогда значения выходной переменной, с учетом принятой базы правил, для регулятора верхнего уровня можно вычислить по формуле среднего взвешенного

$$\lambda^{j} = \frac{\sum_{i=1}^{N_{1}} \delta_{i} c_{i}}{\sum_{i=1}^{N_{1}} \delta_{i}}, \qquad (2.18)$$

где *j* – количество выходов регулятора;

*с*<sub>*i*</sub> – выходные константы.

Степени истинности для базы знаний нечеткого логического регулятора (НЛР) внутреннего каскада, отслеживающего положение по оси абсцисс и ординат, для каждой платформы описывается следующим образом:  $F_g(\alpha_{\phi}^k)$ ,  $G_g(\beta_{\phi}^k)$ , где g – номер правила в базе знаний регулятора второго уровня ( $g = 1...N_2$ );  $\alpha_{\phi}^k$  – те-кущий угол отклонения от нормали по оси абсцисс «k»-й полуплатформы;  $\beta_{\phi}^k$  – текущий угол отклонения от нормали по оси ординат «k»-й полуплатформы.

Аналогично, как и у регулятора верхнего уровня, с помощью операции минимума определим уровни отсечения каждой нечеткой переменной для каждого нечеткого правила

$$\rho_g = \Lambda_g(F_g(\alpha_{\phi}^k), G_g(\beta_{\phi}^k)).$$
(2.19)

Значения выходной переменной, с учетом принятой базы правил, для регулятора внутреннего каскада можно вычислить по формуле среднего взвешенного

$$l_{\nu}^{o} = \frac{\sum_{g=1}^{N_{2}} \rho_{g} s_{g}}{\sum_{g=1}^{N_{2}} \rho_{g}},$$
(2.20)

где о – количество выходов регулятора второго уровня;

*v* – номер полуплатформы;

*s*<sub>*g*</sub> – выходные константы.

Управляющий сигнал для элементов внутреннего каскада второго уровня вложенности, управляющих положением гидроприводов, можно получить по формуле

$$L_i^k = \lambda^j + l_v^o. \tag{2.21}$$

На основе полученного сигнала управления выходное воздействие регулятора второго уровня вложенности можно описать следующим образом:

$$\omega_r = \frac{\sum_{r=1}^{N_3} \min H_r(L^k) \sigma_r}{\sum_{r=1}^{N_3} \min \sigma_r},$$
(2.22)

где r – номер правила в базе нечеткого модуля вложенного каскада ( $r = 1...N_3$ );

Н – функции принадлежности;

 $\sigma_r$  – выходные константы.

Таким образом, итоговое значение управляющего воздействия формируемым многокаскадным нечетким логическим регулятором представляет собой сумму значений выходов вложенного каскада на каждый элемент активной подвески каждой полуплатформы [84].

#### Выводы по второй главе

На основании выведенного математического описания работы гидравлической системы, состоящего из базовых элементов, таких как, гидравлический насос, пропорциональный гидравлический распределитель и гидропривода, была получена имитационная модель одного модуля подвески многоопорной транспортировочной машины. Получены графики переходных характеристик работы гидропривода с применением классического ПИД-закона управления, демонстрирующие реакцию системы на возмущающие воздействия различного рода.

Также в ходе исследования на основании сформулированного алгоритма работы интеллектуальной системы было получено математическое описание многокаскадного нечеткого логического регулятора, состоящего из двух уровней вложенности. Впервые предложена функциональная схема компоновки модулей многокаскадного нечеткого логического регулятора, управляющая подвеской многоопорной машины на основании данных поперечного и продольного изменений положения.

### Глава 3 Синтез и анализ имитационной модели гидравлической системы платформы модульного тралового механизма с многокаскадным нечетким логическим регулятором

На основании описанного в предыдущей главе алгоритма работы многокаскадного нечеткого логического регулятора синтезируем регулятор второго уровня вложенности, отвечающий за управление положением гидравлического привода полуплатформы.

На рисунках 3.1 – 3.2 представлена настройка нечеткого логического регулятора с алгоритмом вывода Такаги-Сугено. При настройке нечеткого логического регулятора, управляющего положением штока, для входной лингвистической переменной используются аппроксимированные функции принадлежности треугольной формы, которые равномерно распределены в диапазоне 0 до 0,5 м (соответствующему максимальному крайнему положению), с пересечением треугольников при значении степени принадлежности 0,5, что удовлетворяет правилу разбиения единицы [46]. Выходные переменные представляют константы – положение золотника гидрораспределителя. Нечеткий логический регулятор настроен таким образом, чтобы поддерживать положение штока гидравлического цилиндра в среднем положении (т.е. 0,25 м).



Рисунок 3.1 – Нечеткий модуль управления положением гидропривода второго уровня вложенности



Рисунок 3.2 – Входная лингвистическая переменная нечеткого логического регулятора второго уровня вложенности

Нечеткая продукционная база знаний регулятора второго уровня вложенности каскада представляет собой перечень из пяти правил и имеет следующий вид:

1 If (input1 is Low) then (output1 = 0,005).

2 If (input1 is LowMid) then (output1 = 0,002).

3 If (input1 is Mid) then (output1 = 0).

4 If (input1 is MidBig) then (output1 = -0,002).

5 If (input1 is Big) then (output1 = -0,005).

На рисунке 3.3 представлена структурная схема гидравлической системы с применением классического ПИД-регулятора и нечеткого логического регулятора для управления положением штока гидропривода.



Рисунок 3.3 – Структурная схема гидравлического привода с ПИД-регулятором и нечеткого логического регулятора второго уровня вложенности



1 – классический ПИД-регулятор; 2 – НЛР второго уровня вложенности

Рисунок 3.4 – Графики переходного процесса по положению гидропривода с подачей возмущающего воздействия трапецеидальной формы

Из графика переходного процесса по положению штока привода, представленного на рисунке 3.4, можно сделать вывод, что нечеткий логический регулятор второго уровня вложенности имеет несущественные отклонения относительно эталонного графика переходного процесса с ПИД-законом управления и сохраняет при этом показатели системы по быстродействию и точности.

При моделировании имитационной модели полуплатформы примем следующие допущения:

- конструктивно трал состоит из одной полуплатформ, которая опирается на четыре гидравлических привода с датчиками линейного перемещения;

- гидравлическая жидкость системы ISO VG 32(ESSO UNIVIS N32);

- каждая полуплатформа представляет собой абсолютно жесткую пластину, не подвергающуюся скручиваниям и изгибам.

Из описанной ранее последовательности работы комплекса составим одну четвертую часть структурной схемы гидравлической системы, стабилизирующей положение груза в пространстве, состоящей из четырех гидравлических цилиндров (рисунок 3.5). На каждом из цилиндров расположен датчик линейного перемещения, обратная связь с которого подается на нечеткий логический регулятор 2-го уровня вложенности (НЛР), который регулирует проходящую жидкость через гидравлический распределитель [79, 89, 93].



Рисунок 3.5 – Структурная схема гидравлической системы полуплатформы трала, состоящая из четырех гидроцилиндров

На рисунке 3.6 представлены графики переходного процесса в гидроприводах с использованием НЛР.



Рисунок 3.6 – Графики переходного процесса в гидроприводах без учета возмущения

На рисунке 3.7 изображены графики переходного процесса в гидроприводах при подаче на систему ступенчатого сигнала с белым шумом, имитирующего неровность дорожного покрытия, а также случайный характер возникновения.



Рисунок 3.7 – Графики переходных процессов при подаче на систему возмущающего воздействия в виде белого шума

На рисунке 3.8 представлен график переходного процесса в гидроприводах при подаче на систему синусоидального сигнала с белым шумом, имитирующего плавное изменение дорожного полотна с возникновением локальных экстремумов. Инверсия сигнала возмущения используется для моделирования изменения дорожного полотна на второй половине полуплатформы.



Рисунок 3.8 – График переходного процесса при подаче на систему возмущающего воздействия в виде синусойды с белым шумом

Данная система позволяет снизить влияние неровности дороги на положение платформы, но она не решает ряд проблем, связанных с кривизной дорожного покрытия и высокой парусностью. Для снижения возможных рисков, связанных со стабилизацией трала относительно горизонтального положения и приводящих к его кренам, введем в систему каскад НЛР первого уровня вложенности, который позволит снизить представленные негативные явления. Как было отмечено ранее, применение нечеткой логики в гидравлической системе может позволить снизить возможные риски, связанные с неровностью дорожного покрытия, скоростью бокового ветра и выходом из строя элементов подвески.

Основная задача первого уровня вложенности каскада НЛР заключается в стабилизации платформы в пространстве по горизонтали путем подачи управляющего сигнала на нечеткие логические контроллеры внутреннего каскада (далее модули). Положение платформы изменяется относительно двух осей: *X* и *Y* (см. рисунок 1.8).

Опираясь на принятые ранее допущения, рассмотрим модель полуплатформы, состоящую из четырех гидроцилиндров. Данная модель упрощает математическое описание системы, но позволяет сохранить качественные характеристики [92].

Полуплатформа представляет собой абсолютно жесткую пластину, не подвергающуюся скручиваниям и изгибам. Размер платформы – 4200×3000 мм.

Максимальный угол отклонения платформы по оси *X* достигается при условии, что гидроцилиндры 21 и 22 выдвинуты на 0,5 м, а гидроцилиндры 11 и 12 полностью втянуты. Максимальный угол отклонения платформы по оси *Y* достигается при условии, что гидроцилиндры 11 и 21 выдвинуты на 0,5 м, а гидроцилиндры 12 и 22 полностью втянуты (рисунок 3.9).



Рисунок 3.9 – Расчет максимального угла отклонения платформы по оси абсцисс и ординат

При принятых габаритных размерах тралового механизма и крайних состояниях положения подвесной системы угол наклона по оси абсцисс не должен превышать 6°, а по оси ординат – 9°, исходя из расчета угла треугольника, приведенного на рисунке 3.9 [9].

Моделирование данных, полученных от датчика горизонтального положения, осуществляется путем подачи возмущающего воздействия на две координаты системы – X и Y. Диапазоны значений координат  $X = \pm 6^{\circ}$  и  $Y = \pm 9^{\circ}$ , обусловлены максимально возможным креном платформы относительно ее горизонтального положения [92].

Нечеткий логический регулятор каскада второго уровня вложенности реализован в двух вариантах:

1) нечеткий логический регулятор, формирующий уставки внутренних модулей за счет суммирования значений сигналов управления;

2) нечеткий логический регулятор, реализующий классификацию внутренних модулей по пропорциональному принципу с применением оператора умножения.

На рисунке 3.10 представлена структурная схема гидравлической системы полуплатформы с нечетким логическим регулятором первого уровня вложенности, формирующим уставки для вложенного каскада второго уровня за счет суммирования значений сигналов управления.

Настройка каскада первого уровня вложенности нечеткого логического регулятора с алгоритмом вывода Такаги-Сугено для управления гидравлической системой многосоставного тралового механизма, формирующего уставки внутренних модулей за счет суммирования значений сигналов управления, представлена на рисунках 3.11 - 3.13. Входные лингвистические переменные «*X*» и «*Y*» – аппроксимированные функции принадлежности треугольной формы, состоят из пяти элементов терм-множеств, которые равномерно распределены в диапазоне  $X = \pm 6^{\circ}$  и  $Y = \pm 9^{\circ}$ , определяющих горизонтальное отклонение транспортировочного аппарата. Пересечение треугольников при значении степени принадлежности 0,5 удовлетворяет правилу разбиения единицы. Сжатие нечеткого терма null обусловлено повышением точности интеллектуальной системы управления при ми-

нимизации ошибки системы по положению трала в пространстве. Выходные переменные регулятора представляют собой константы – сигналы управления на модули второго уровня вложенности каскада.



Рисунок 3.10 – Структурная схема гидравлической системы полуплатформы с нечетким логическим регулятором, формирующим уставки для модулей
1-го уровня вложенности за счет суммирования значений сигналов управления



Рисунок 3.11 – Нечеткий модуль управления первого уровня вложенности каскада, формирующий уставки внутренних модулей за счет суммирования значений сигналов управления



Рисунок 3.12 – Входная лингвистическая переменная «У»



Рисунок 3.13 – Входная лингвистическая переменная «Х»

База знаний каскада первого уровня вложенности НЛР состоит из 25 правил и имеет следующую форму представления:

1 If (Y is null) and (X is null) then (Out11 = 0) (Out12 = 0) (Out21 = 0) (Out22 = 0).

2 If (Y is low\_Incline) and (X is null) then (Out11 = -0,25) (Out12 = 0) (Out21 = -0,25) (Out22 = 0).

3 If (Y is low\_MidIncline) and (X is null) then (Out11 = -0,125) (Out12 = 0) (Out21 = -0,125) (Out22 = 0).

4 If (Y is high\_MidIncline) and (X is null) then (Out11 = 0) (Out12 = -0,125) (Out21 = 0) (Out22 = -0,125). 5 If (Y is high\_Incline) and (X is null) then (Out11 = 0) (Out12 = -0,25) (Out21 = 0) (Out22 = -0,25).

6 If (Y is null) and (X is low\_Incline) then (Out11 = -0,25) (Out12 = -0,25) (Out21 = 0) (Out22 = 0).

7 If (Y is null) and (X is low\_MidIncline) then (Out11 = -0,125) (Out12 = -0,125) (Out21 = 0) (Out22 = 0).

8 If (Y is null) and (X is highMidIncline) then (Out11 = 0) (Out12 = 0) (Out21 = = -0,125) (Out22 = -0,125).

9 If (Y is null) and (X is high\_Incline) then (Out11 = 0) (Out12 = 0) (Out21 = -0,25) (Out22 = -0,25).

10 If (Y is high\_Incline) and (X is high\_Incline) then (Out11 = 0,25) (Out12 = 0) (Out21 = 0) (Out22 = -0,25).

11 If (Y is high\_Incline) and (X is high\_MidIncline) then (Out11 = 0,25) (Out12 = = 0) (Out21 = 0) (Out22 = -0,25).

12 If (Y is high\_MidIncline) and (X is high\_Incline) then (Out11 = 0,25) (Out12 = = 0) (Out21 = 0) (Out22 = -0,25).

13 If (Y is high\_MidIncline) and (X is high\_MidIncline) then (Out11 = 0,125) (Out12 = 0) (Out21 = 0) (Out22 = -0,125).

14 If (Y is high\_Incline) and (X is low\_Incline) then (Out11 = 0) (Out12 = -0,25) (Out21 = 0,25) (Out22 = 0).

15 If (Y is high\_MidIncline) and (X is low\_Incline) then (Out11 = 0) (Out12 = -0,25) (Out21 = 0,25) (Out22 = 0).

16 If (Y is high\_Incline) and (X is low\_MidIncline) then (Out11 = 0) (Out12 = -0,25) (Out21 = 0,25) (Out22 = 0).

17 If (Y is high\_MidIncline) and (X is low\_MidIncline) then (Out11 = 0) (Out12 = -0,125) (Out21 = 0,125) (Out22 = 0).

18 If (Y is low\_Incline) and (X is high\_Incline) then (Out11 = 0) (Out12 = 0,25) (Out21 = -0,25) (Out22 = 0).

19 If (Y is low\_MidIncline) and (X is high\_Incline) then (Out11 = 0) (Out12 = = 0,25) (Out21 = -0,25) (Out22 = 0).

20 If (Y is low\_Incline) and (X is high\_MidIncline) then (Out11 = 0) (Out12 = = 0,25) (Out21 = -0,25) (Out22 = 0).

21 If (Y is low\_MidIncline) and (X is high\_MidIncline) then (Out11 = 0) (Out12 = 0,125) (Out21 = -0,125) (Out22 = 0).

22 If (Y is low\_Incline) and (X is low\_Incline) then (Out11 = -0.25) (Out12 = 0) (Out21 = 0) (Out22 = 0.25).

23 If (Y is low\_MidIncline) and (X is low\_Incline) then (Out11 = -0,25) (Out12 = = 0) (Out21 = 0) (Out22 = 0,25).

24 If (Y is low\_Incline) and (X is low\_MidIncline) then (Out11 = -0,25) (Out12 = = 0) (Out21 = 0) (Out22 = 0,25).

25 If (Y is low\_MidIncline) and (X is low\_MidIncline) then (Out11 = -0,125) (Out12 = 0) (Out21 = 0) (Out22 = 0,125).

На рисунке 3.14 представлен график переходного процесса в гидроприводах с использованием нечеткого модуля управления первого уровня вложенности каскада, формирующего уставки внутренних модулей за счет суммирования значений сигналов управления [110].

Графики переходных процессов в гидроприводах при подаче на систему возмущающих воздействий в виде синусоидального сигнала и синусоидального сигнала с белым шумом позволят смоделировать неровность дорожного покрытия, а также локальные экстремумы (рисунки 3.15 – 3.16).



Рисунок 3.14 – Графики переходных процессов в гидроприводах при подаче на систему управляющего воздействия без учета возмущения



Рисунок 3.15 – Графики переходных процессов в гидроприводах при подаче на систему возмущающего воздействия в виде синусоидального сигнала

На рисунке 3.16 представлен график переходного процесса в гидроприводах с использованием нечеткого модуля управления первого уровня вложенности каскада, формирующего уставки для модулей второго уровня вложенности каскада за счет суммирования значений сигналов управления при подаче на систему

68

синусоидального сигнала с белым шумом, имитирующего неровность дорожного покрытия.



Рисунок 3.16 – Графики переходных процессов в гидроприводах при подаче на систему возмущающего воздействия в виде синусоидального сигнала с белым шумом

Рассмотрим графики переходных процессов в гидроцилиндрах после работы нечеткого логического регулятора каскада первого уровня вложенности:

1 Временной период I (время с 5 до 10 с). Во временном периоде с 5 до 10 с на вход нечеткого логического регулятора первого уровня вложенности поступают сигналы от гироскопического датчика, представленного в виде ступенчатой функции со следующими показателями:  $X = 0^{\circ}$ ,  $Y = +9^{\circ}$ , указывающими на отклонение платформы от горизонтального положения на определенный угол. Данные датчика представляются интеллектуальной системой как «появление крена на транспортировочном аппарате относительно оси *Y*». Реагируя на полученные значения от гироскопического датчика, интеллектуальная система синтезирует управляющие выходные воздействия для каскада второго уровня вложенности, отвечающего за работу пропорциональных гидравлических распределителей, выдвигая гидроприводы 12 и 22 (см. рисунок 3.16), которые корректируют положение транспортного аппарата в пространстве.

2 Временной период II (время с 15 до 20 с). Во временном периоде с 15 до 20 с на вход нечеткого логического регулятора первого уровня вложенности поступают сигналы от гироскопического датчика, представленного в виде ступенчатой функции со следующими показателями:  $X = -6^{\circ}$ ,  $Y = 0^{\circ}$ , указывающими на отклонение платформы от горизонтального положения на определенный угол. Данные датчика представляются интеллектуальной системой как «появление крена на транспортировочном аппарате относительно оси X». Реагируя на полученные значения от гироскопического датчика, интеллектуальная система синтезирует управляющие выходные воздействия для каскада второго уровня вложенности, отвечающего за работу пропорциональных гидравлических распределителей, выдвигая гидроприводы 11 и 12 (см. рисунок 3.16), которые корректируют положение транспортного аппарата в пространстве.

3 Временной период III (время с 25 до 30 с). Во временном периоде с 25 до 30 с на вход нечеткого логического регулятора первого уровня вложенности поступают сигналы от гироскопического датчика, представленного в виде ступенчатой функции со следующими показателями:  $X = +3^{\circ}$ ,  $Y = -4,5^{\circ}$ , указывающими на отклонение платформы от горизонтального положения на определенный угол. Данные датчика представляются интеллектуальной системой как «появление крена на транспортировочном аппарате относительно осей X и Y». Реагируя на полученные значения от гироскопического датчика, интеллектуальная система синтезирует управляющие выходные воздействия для каскада второго уровня вложенности, отвечающего за работу пропорциональных гидравлических распределителей, выдвигая гидропривод 21 и втягивая гидропривод 12 (см. рисунок 3.16), корректируя положение транспортного аппарата в пространстве.

4 Временной период IV (время с 35 до 40 с). Во временном периоде с 35 до 40 с на вход нечеткого логического регулятора первого уровня вложенности поступают сигналы от гироскопического датчика, представленного в виде ступенчатой функции со следующими показателями:  $X = -5^{\circ}$ ,  $Y = -6^{\circ}$ , указывающими на отклонение платформы от горизонтального положения на определенный угол. Данные датчика представляются интеллектуальной системой как «появление крена на транспортировочном аппарате относительно осей X и Y». Реагируя на полученные значения от гироскопического датчика, интеллектуальная система синтезирует управляющие выходные воздействия для каскада второго уровня вложенности, отвечающего за работу пропорциональных гидравлических распределителей, выдвигая гидропривод 11 и втягивая гидропривод 22 (см. рисунок 3.16), корректируя положение транспортного аппарата в пространстве [79].

На рисунке 3.17 представлена структурная схема гидравлической системы полуплатформы с нечетким логическим регулятором первого уровня вложенности каскада, реализующим классификацию модулей второго уровня вложенности каскада по пропорциональному принципу с применением оператора умножения.



Рисунок 3.17 – Структурная схема гидравлической системы полуплатформы с нечетким логическим регулятором, реализующим классификацию модулей 2-го уровня вложенности каскада по пропорциональному принципу с применением оператора умножения

На рисунках 3.18 – 3.20 представлена настройка нечеткого логического регулятора первого уровня вложенности каскада с алгоритмом вывода Такаги-Сугено. Входные лингвистические переменные «Х» и «Y» состоят из пяти элементов в базовом терм-множестве и имеют треугольную функцию принадлежности, определяющей горизонтальное отклонение транспортной платформы. Пересечение треугольников при значении степени принадлежности 0,5 удовлетворяет правилу разбиения единицы. Выходные переменные регулятора представляют константы – сигнал управления на модули внутреннего каскада.


Рисунок 3.18 – Нечеткий логический регулятор первого уровня вложенности каскада, реализующий классификацию модулей второго уровня вложенности по пропорциональному принципу с применением оператора умножения



Рисунок 3.19 – Входная лингвистическая переменная «У» регулятора первого уровня вложенности, реализующего классификацию модулей второго уровня вложенности по пропорциональному принципу с применением оператора умножения



Рисунок 3.20 – Входная лингвистическая переменная «Х» регулятора первого уровня вложенности, реализующего классификацию модулей второго уровня вложенности по пропорциональному принципу с применением оператора

#### умножения

Нечеткая продукционная база знаний регулятора первого уровня вложенности каскада представляет собой перечень из 25 правил и имеет следующий вид:

1 If (Y is null) and (X is null) then (Out11 = 1) (Out12 = 1) (Out21 = 1) (Out22 = 1).

2 If (Y is lowIncline) and (X is null) then (Out11 = 0,5) (Out12 = 1) (Out21 = 0,5) (Out22 = 1).

3 If (Y is lowMidIncline) and (X is null) then (Out11 = 0,75) (Out12 = 1) (Out21 = 0,75) (Out22 = 1).

4 If (Y is highMidIncline) and (X is null) then (Out11 = 1) (Out12 = 0,75) (Out21 = 1) (Out22 = 0,75).

5 If (Y is highIncline) and (X is null) then (Out11 = 1) (Out12 = 0,5) (Out21 = 1)(Out22 = 0,5).

6 If (Y is null) and (X is lowIncline) then (Out11 = 0,5) (Out12 = 0,5) (Out21 = = 1) (Out22 = 1).

7 If (Y is null) and (X is lowMidIncline) then (Out11 = 0,75) (Out12 = 0,75) (Out21 = 1) (Out22 = 1). 8 If (Y is null) and (X is highMidIncline) then (Out11 = 1) (Out12 = 1) (Out21 = 0,75) (Out22 = 0,75).

9 If (Y is null) and (X is highIncline) then (Out11 = 1) (Out12 = 1) (Out21 = 0,5) (Out22 = 0,5).

10 If (Y is lowIncline) and (X is lowIncline) then (Out11 = 0,5) (Out12 = 1) (Out21 = 1) (Out22 = 50).

11 If (Y is lowIncline) and (X is lowMidIncline) then (Out11 = 0,5) (Out12 = 1) (Out21 = 1) (Out22 = 50).

12 If (Y is lowMidIncline) and (X is lowIncline) then (Out11 = 0,5) (Out12 = 1) (Out21 = 1) (Out22 = 50).

13 If (Y is lowMidIncline) and (X is lowMidIncline) then (Out11 = 0,75) (Out12 = 1) (Out21 = 1) (Out22 = 25).

14 If (Y is highIncline) and (X is highIncline) then (Out11 = 50) (Out12 = 1) (Out21 = 1) (Out22 = 0,5).

15 If (Y is highIncline) and (X is highMidIncline) then (Out11 = 50) (Out12 = 1) (Out21 = 1) (Out22 = 0,5).

16 If (Y is highMidIncline) and (X is highIncline) then (Out11 = 50) (Out12 = 1) (Out21 = 1) (Out22 = 0,5).

17 If (Y is highMidIncline) and (X есть highMidIncline) then (Out11 = 25) (Out12 = 1) (Out21 = 1) (Out22 = 0,75).

18 If (Y is highIncline) and (X is lowIncline) then (Out11 = 1) (Out12 = 0,5) (Out21 = 1) (Out22 = 50).

19 If (Y is highIncline) and (X is lowMidIncline) then (Out11 = 1) (Out12 = 0,5) (Out21 = 1) (Out22 = 50).

20 If (Y is highMidIncline) and (X is lowIncline) then (Out11 = 1) (Out12 = 0,5) (Out21 = 1) (Out22 = 50).

21 If (Y is highMidIncline) and (X is lowMidIncline) then (Out11 = 1) (Out12 = = 0,75) (Out21 = 1) (Out22 = 25).

22 If (Y is lowIncline) and (X is highIncline) then (Out11 = 1) (Out12 = 50) (Out21 = 1) (Out22 = 0,5). 23 If (Y is lowIncline) and (X is highMidIncline) then (Out11 = 1) (Out12 = 50) (Out21 = 1) (Out22 = 0,5).

24 If (Y is lowMidIncline) and (X is highIncline) then (Out11 = 1) (Out12 = 50) (Out21 = 1) (Out22 = 0,5).

25 If (Y is lowMidIncline) and (X is highMidIncline) then (Out11 = 1) (Out12 = = 25) (Out21 = 1) (Out22 = 0,75).

На рисунке 3.21 представлен график переходного процесса в гидроприводах с использованием нечеткого логического регулятора, реализующего классификацию модулей второго уровня вложенности по пропорциональному принципу с применением оператора умножения.

На рисунках 3.22 – 3.23 представлены графики переходного процесса в гидроприводах с использованием нечеткого модуля управления каскада первого уровня вложенности, реализующего классификацию модулей второго уровня вложенности по пропорциональному принципу с применением оператора умножения при подаче на систему синусоидального сигнала с белым шумом, имитирующего неровность дорожного покрытия.



Рисунок 3.21 – Графики переходных процессов в гидроприводах при подаче на систему управляющего воздействия без учета возмущения



Рисунок 3.22 – Графики переходных процессов в гидроприводах при подаче на систему возмущающего воздействия в виде синусоидального сигнала



Рисунок 3.23 – Графики переходных процессов в гидроприводах при подаче на систему возмущающего воздействия в виде синусоидального сигнала с белым шумом

1 Временной период I (интервал 5 до 10 с). Во временной период от 5 до 10 с на вход НЛР первого уровня вложенности поступают сигналы от датчика углового положения, представленного в виде ступенчатой функции со следующим набором параметров:  $X = +0^\circ$ ,  $Y = +9^\circ$ , указывающих на отклонение платформы от

79

горизонтального положения на определенный угол. Данные датчика представляются интеллектуальной системой как «появление крена на транспортировочном аппарате относительно оси *Y*». Реагируя на полученную информацию от датчика углового положения, интеллектуальная система синтезирует управляющие воздействия для модулей второго уровня вложенности каскада, отвечающего за изменения положений золотников пропорциональных гидравлических распределителей, выдвигая гидроприводы 12 и 22, которые корректируют положение многоопорной машины в пространстве.

2 Временной период II (интервал 15 до 20 с). Во временном периоде от 15 до 20 с на вход НЛР первого уровня вложенности поступают сигналы от датчика углового положения, представленного в виде ступенчатой функции со следующим набором параметров:  $X = -6^{\circ}$ ,  $Y = 0^{\circ}$ , указывающих на отклонение платформы от горизонтального положения на определенный угол. Данные датчика представляются интеллектуальной системой как «появление крена на транспортировочном аппарате относительно оси X». Реагируя на полученную информацию от датчика углового положения, интеллектуальная система синтезирует управляющие воздействия для модулей второго уровня вложенности каскада, отвечающего за изменения положений золотников пропорциональных гидравлических распределителей, выдвигая гидроприводы 11 и 12, которые корректируют положение многоопорной машины в пространстве.

3 Временной период III (период с 25 до 30 и 35 до 40 с). При возникновении отклонений по оси абсцисс  $X = + 3^{\circ}$  и ординат  $Y = -4,5^{\circ}$  в момент времени 25 с и  $X = -5^{\circ}$ , Y = -6 в момент времени 35 с регулятор работает некорректно. Такое поведение системы обусловлено тем, что формирование уставок для регуляторов вложенного каскада производится по пропорциональному принципу с применением оператора произведения. При возникновении скачкообразных отклонений нечеткий логический регулятор внешнего каскада синтезирует сигнал управления, который кратно увеличивает входной сигнал внутренних модулей, что приводит к некорректной реакции системы.

# 3.1 Определение ограничений, накладываемых на систему управления четырехсекционной транспортной платформой

Для стабилизации транспортного трала, состоящего из четырехсекционной транспортной платформы на одном уровне в условиях изменения наклона дорожного полотна, необходимо рассчитать все ограничения, накладываемые на систему для ее корректной эксплуатации.

Примем следующие допущения:

 конструктивно трал состоит из четырех полуплатформ, каждая из которых опирается на четыре гидравлических привода с датчиками линейного перемещения;

- каждая полуплатформа представляет собой абсолютно жесткую пластину, не подвергающуюся скручиваниям и изгибам;

- целостность и твердость шин являются абсолютными;

- расстояние между соседними платформами (зазоры) настолько малы, что ими можно пренебречь.

Задача системы управления, реализованной на основе нечеткого логического регулятора, формирующего уставки внутренних модулей управления за счет суммирования значений сигналов регулирования, заключается в удержании всех четырех полуплатформ на одном уровне таким образом, чтобы при движении по дороге, имеющей наклон, не образовывалось вертикальное смещение платформ относительно друг друга (рисунки 3.24 – 3.25).



Рисунок 3.24 – Разница в высоте платформ



Рисунок 3.25 – Стабилизация платформ по высоте

В качестве датчика для определения рассогласования между платформами в вертикальной проекции может быть использован закрепленный на одной из полуплатформ и соединенный с соседней с помощью тяги.

Опираясь на принятые ранее допущения, рассмотрим модель трала, состоящего из четырех полуплатформ. Каждая полуплатформа состоит из четырех гидроцилиндров. Данная модель упрощает математическое описание системы, но позволяет сохранить качественные характеристики.

Полуплатформа представляет собой абсолютно жесткую пластину, не подвергающуюся скручиваниям и изгибам. Размер платформы – 4200×3000 мм. Следовательно, модульный траловый механизм, состоящий из четырех полуплатформ, имеет размеры 12 800×3000 мм [84].

Условие для преодоления максимального угла наклона дорожного полотна определяется из условия, что гидроцилиндры 11 и 12 полностью втянуты, а гидроцилиндры 43 и 44 полностью выпущены (рисунок 3.26).

Рассчитаем максимальный угол наклона дорожного полотна, при котором система управления тралом позволяет удерживать все четыре полуплатформы на одном уровне с учетом, что максимальный угол наклона дорожного полотна достигается при условии, что гидроцилиндры 11 и 12 полностью втянуты, а гидроцилиндры 43 и 44 полностью выпущены (см. рисунок 3.26).

Исходя из того, что модульный траловый механизм, состоящий из четырех последовательно соединенных модулей, имеет длину 12 800 мм и ход подвесной системы составляет 500 мм, угол наклона дорожного полотна не должен превышать 2.23° на основе расчета угла треугольника, приведенного на рисунке 3.27 [9].

82



Рисунок 3.26 – Функциональная схема

гидравлической системы трала





дорожного полотна

При движении секции, состоящей из нескольких полуплатформ по наклонной поверхности, неизбежно образовывается вертикальное смещение платформ (см. рисунок 3.24). Для создания системы управления на базе НЛР, которая позволит стабилизировать платформы, необходимо сформировать продукционную базу правил. Для этого рассмотрим возможные варианты положения полуплатформ во время движения по дорожному полотну без наклона, с наклоном в подъем и с наклоном на спуске.

#### Движение трала по ровной поверхности

#### **Положение 1** (рисунок 3.28)

Все платформы располагаются на одном уровне. Сигналы с датчиков положения платформ равны нулю, при этом штоки гидроцилиндров 11 и 44 находятся в среднем положении. При данных входных значениях НЛР формирует управляющее воздействие, равное нулю, во избежание разбалансировки системы.



Рисунок 3.28 – Положение платформы при движении по ровной поверхности

#### Положение 2 (см. рисунок 3.28)

Все платформы располагаются на одном уровне. Сигналы с датчиков положения платформ равны нулю, при этом штоки гидроцилиндров 11 и 44 выдвинуты выше среднего положения. При данных входных значениях НЛР формирует управляющее воздействие – втянуть все штоки гидроцилиндров.

#### Положение 3 (см. рисунок 3.28)

Все платформы располагаются на одном уровне. Сигналы с датчиков положения платформ равны нулю, при этом штоки гидроцилиндров 11 и 44 выдвинуты ниже среднего положения. При данных входных значениях НЛР формирует управляющее воздействие – выдвинуть все штоки гидроцилиндров.

## Движение трала в подъем

#### Положение 4 (рисунок 3.29)

Полуплатформа 1 выше уровня остальных полуплатформ, полуплатформы 2, 3 и 4 располагаются на одном уровне. Сигнал с датчика 1-2 больше нуля, с датчиков 2-3 и 3-4 равен нулю, при этом шток гидроцилиндра 11 втянут не до упора. При данных входных значениях НЛР формирует управляющее воздействие – втянуть все штоки гидроцилиндров полуплатформы 1, не изменять положение гидроцилиндров полуплатформ 2, 3 и 4.



Рисунок 3.29 – Положение платформы 1 выше уровня платформ 2-4 при движении в подъем

## Положение 5 (см. рисунок 3.29)

Полуплатформа 1 выше уровня остальных полуплатформ, полуплатформы 2, 3 и 4 располагаются на одном уровне. Сигнал с датчика 1-2 больше нуля, с датчиков 2-3 и 3-4 равен нулю, при этом шток гидроцилиндра 11 втянут до упора. При данных входных значениях НЛР формирует управляющее воздействие – выдвинуть все штоки гидроцилиндров полуплатформ 2, 3, 4, не изменять положение гидроцилиндров полуплатформы 1.

## Положение 6 (рисунок 3.30)

Полуплатформа 1 ниже уровня остальных полуплатформ, полуплатформы 2, 3 и 4 располагаются на одном уровне. Сигнал с датчика 1-2 меньше нуля, с датчиков 2-3 и 3-4 равны нулю, при этом шток гидроцилиндра 11 выдвинут не до упора. При данных входных значениях НЛР формирует управляющее воздействие – выдвинуть все штоки гидроцилиндров полуплатформы 1, не изменять положение гидроцилиндров полуплатформ 2 и 3, 4.



Рисунок 3.30 – Положение платформы 1 ниже уровня платформ 2-4 при движении в подъем

Положение 7 (см. рисунок 3.30)

Полуплатформа 1 ниже уровня остальных полуплатформ, полуплатформы 2, 3 и 4 располагаются на одном уровне. Сигнал с датчика 1-2 меньше нуля, с датчиков 2-3 и 3-4 равны нулю, при этом шток гидроцилиндра 11 выдвинут до упора. При данных входных значениях НЛР формирует управляющее воздействие – втянуть все штоки гидроцилиндров полуплатформ 2, 3, 4, не изменять положение гидроцилиндров полуплатформы 1.

Положение 8 (рисунок 3.31)

Полуплатформа 2 выше уровня полуплатформ 1, 3 и 4, полуплатформы 1, 3 и 4 располагаются на одном уровне. Сигнал с датчика 1-2 меньше нуля, с датчика 2-3 больше нуля, с датчика 3-4 равен нулю, при этом положение штоков гидроцилиндров не учитывается. При данных входных значениях НЛР формирует управляющее воздействие – втянуть все штоки гидроцилиндров полуплатформы 2, не изменять положение гидроцилиндров полуплатформ 1, 3 и 4.



Рисунок 3.31 – Положение платформы 2 выше уровня платформ 1,3.4 при движении в подъем

#### **Положение 9** (рисунок 3.32)

Полуплатформа 2 ниже уровня полуплатформ 1, 3 и 4, полуплатформы 1, 3 и 4 располагаются на одном уровне. Сигнал с датчика 1-2 больше нуля, с датчика 2-3 меньше нуля, с датчика 3-4 равен нулю, при этом положение штоков гидроцилиндров не учитывается. При данных входных значениях НЛР формирует управляющее воздействие – выдвинуть все штоки гидроцилиндров полуплатформы 2, не изменять положение гидроцилиндров полуплатформ 1, 3 и 4.



Рисунок 3.32 – Положение платформы 2 ниже платформ 1, 3, 4 при движении в подъем

## Положение 10 (рисунок 3.33)

Полуплатформы 1, 2 и 4 располагаются на одном уровне, полуплатформа 3 выше уровня полуплатформ 1, 2 и 4. Сигнал с датчика 1-2 равен нулю, с датчика 2-3 меньше нуля, с датчика 3-4 больше нуля, при этом положение штоков гидроцилиндров не учитывается. При данных входных значениях НЛР формирует управляющее воздействие – втянуть все штоки гидроцилиндров полуплатформы 3, не изменять положение гидроцилиндров полуплатформ 1, 2 и 4.



Рисунок 3.33 – Положение платформы 3 выше платформ 1, 2, 4 при движении в подъем

#### **Положение 11** (рисунок 3.34)

Полуплатформы 1, 2 и 4 располагаются на одном уровне, полуплатформа 3 ниже уровня полуплатформ 1, 2 и 4. Сигнал с датчика 1-2 равен нулю, с датчика 2-3 больше нуля, с датчика 3-4 меньше нуля, при этом положение штоков гидроцилиндров не учитывается. При данных входных значениях НЛР формирует управляющее воздействие – выдвинуть все штоки гидроцилиндров полуплатформы 3, не изменять положение гидроцилиндров полуплатформ 1, 2 и 4.



Рисунок 3.34 – Положение платформы 3 ниже платформ 1, 2, 4 при движении в подъем

## Положение 12 (рисунок 3.35)

Полуплатформы 1, 2 и 3 располагаются на одном уровне, полуплатформа 4 выше уровня полуплатформ 1, 2 и 3. Сигнал с датчиков 1-2 и 2-3 равен нулю, с датчика 3-4 меньше нуля, при этом шток гидроцилиндра 44 втянут не до упора. При данных входных значениях НЛР формирует управляющее воздействие – втянуть все штоки гидроцилиндров полуплатформы 4, не изменять положение гидроцилиндров полуплатформ 1, 2 и 3.



Рисунок 3.35 – Положение платформы 4 выше платформ 1, 2, 3 при движении в подъем

#### Положение 13 (см. рисунок 3.35)

Полуплатформы 1, 2 и 3 располагаются на одном уровне, полуплатформа 4 выше уровня полуплатформ 1, 2 и 3. Сигнал с датчиков 1-2 и 2-3 равен нулю, с датчика 3-4 меньше нуля, при этом шток гидроцилиндра 44 втянут до упора. При данных входных значениях НЛР формирует управляющее воздействие – выдвинуть все штоки гидроцилиндров полуплатформ 1, 2 и 3, не изменять положение гидроцилиндров полуплатформы 4.

## Положение 14 (рисунок 3.36)

Полуплатформы 1, 2 и 3 располагаются на одном уровне, полуплатформа 4 ниже уровня полуплатформ 1, 2 и 3. Сигнал с датчиков 1-2 и 2-3 равен нулю, с датчика 3-4 больше нуля, при этом шток гидроцилиндра 44 выдвинут не до упора. При данных входных значениях НЛР формирует управляющее воздействие – выдвинуть все штоки гидроцилиндров полуплатформы 4, не изменять положение гидроцилиндров полуплатформ 1, 2 и 3.



Рисунок 3.36 – Положение платформы 4 ниже платформ 1, 2, 3 при движении в подъем

## Положение 15 (см. рисунок 3.36)

Полуплатформы 1, 2 и 3 располагаются на одном уровне, полуплатформа 4 ниже уровня полуплатформ 1, 2 и 3. Сигнал с датчиков 1-2 и 2-3 равен нулю, с датчика 3-4 больше нуля, при этом шток гидроцилиндра 44 выдвинут до упора. При данных входных значениях НЛР формирует управляющее воздействие – втя-

нуть все штоки гидроцилиндров полуплатформ 1, 2 и 3, не изменять положение гидроцилиндров полуплатформы 4.

## Положение 16 (рисунок 3.37)

Полуплатформы 1 и 2 располагаются на одном уровне и одновременно выше уровня полуплатформ 3 и 4, полуплатформы 3 и 4 располагаются на одном уровне. Сигнал с датчиков 1-2 и 3-4 равен нулю, с датчика 2-3 больше нуля, при этом шток гидроцилиндра 44 выдвинут не до упора, шток гидроцилиндра 11 втянут не до упора. При данных входных значениях НЛР формирует управляющее воздействие – втянуть все штоки гидроцилиндров полуплатформ 1 и 2, выдвинуть все штоки гидроцилиндров полуплатформ 3 и 4.



Рисунок 3.37 – Положение платформ 1 и 2 выше платформ 3 и 4 при движении в подъем

#### Положение 17 (см. рисунок 3.37)

Полуплатформы 1 и 2 располагаются на одном уровне и одновременно выше уровня полуплатформ 3 и 4, полуплатформы 3 и 4 располагаются на одном уровне. Сигнал с датчиков 1-2 и 3-4 равен нулю, с датчика 2-3 больше нуля, при этом шток гидроцилиндра 44 выдвинут не до упора, шток гидроцилиндра 11 втянут до упора. При данных входных значениях НЛР формирует управляющее воздействие – не изменять положение гидроцилиндров полуплатформ 1 и 2, выдвинуть все штоки гидроцилиндров полуплатформ 3 и 4.

### Положение 18 (см. рисунок 3.37)

Полуплатформы 1 и 2 располагаются на одном уровне и одновременно выше уровня полуплатформ 3 и 4, полуплатформы 3 и 4 располагаются на одном уровне. Сигнал с датчиков 1-2 и 3-4 равен нулю, с датчика 2-3 больше нуля, при этом шток гидроцилиндра 44 выдвинут до упора, шток гидроцилиндра 11 втянут не до упора. При данных входных значениях НЛР формирует управляющее воздействие – втянуть все штоки гидроцилиндров полуплатформ 1 и 2, не изменять положение гидроцилиндров полуплатформ 3 и 4.

#### **Положение 19** (рисунок 3.38)

Полуплатформы 1 и 2 располагаются на одном уровне и одновременно выше уровня полуплатформ 3 и 4, полуплатформы 3 и 4 располагаются на одном уровне. Сигнал с датчиков 1-2 и 3-4 равен нулю, с датчика 2-3 меньше нуля, при этом шток гидроцилиндра 44 втянут не до упора, шток гидроцилиндра 11 выдвинут не до упора. При данных входных значениях НЛР формирует управляющее воздействие – выдвинуть все штоки гидроцилиндров полуплатформ 1 и 2, втянуть все штоки гидроцилиндров полуплатформ 3 и 4.



Рисунок 3.38 – Положение платформ 1 и 2 ниже платформ 3 и 4 при движении в подъем

#### Положение 20 (см. рисунок 3.38)

Полуплатформы 1 и 2 располагаются на одном уровне и одновременно выше уровня полуплатформ 3 и 4, полуплатформы 3 и 4 располагаются на одном уровне. Сигнал с датчиков 1-2 и 3-4 равен нулю, с датчика 2-3 меньше нуля, при этом шток гидроцилиндра 44 втянут до упора, шток гидроцилиндра 11 выдвинут не до упора. При данных входных значениях НЛР формирует управляющее воздействие – выдвинуть все штоки гидроцилиндров полуплатформ 1 и 2, не изменять положение гидроцилиндров полуплатформ 3 и 4.

### Положение 21 (см. рисунок 3.38)

Полуплатформы 1 и 2 располагаются на одном уровне и одновременно выше уровня полуплатформ 3 и 4, полуплатформы 3 и 4 располагаются на одном уровне. Сигнал с датчиков 1-2 и 3-4 равен нулю, с датчика 2-3 меньше нуля, при этом шток гидроцилиндра 44 втянут не до упора, шток гидроцилиндра 11 выдвинут до упора. При данных входных значениях НЛР формирует управляющее воздействие – не изменять положение гидроцилиндров полуплатформ 1 и 2, втянуть все штоки гидроцилиндров полуплатформ 3 и 4.

#### Движение трала на спуске

Положение 22 (рисунок 3.39) идентично положениям 6, 7 (см. рисунок 3.30).



Рисунок 3.39 – Положение платформы 1 ниже платформ 2, 3, 4 при движении на спуске

Положение 23 (рисунок 3.40) идентично положениям 4, 5 (см. рисунок 3.29).



Рисунок 3.40 – Положение платформы 1 выше платформ 2, 3, 4 при движении на спуске

Положение 24 (рисунок 3.41) идентично положению 9 (см. рисунок 3.32).



Рисунок 3.41 – Положение платформы 2 ниже платформ 1, 3, 4 при движении на спуске

## Положение 25 (рисунок 3.42) идентично положению 8 (см. рисунок 3.31).



Рисунок 3.42 – Положение платформы 2 выше платформ 1, 3, 4 при движении на спуске

Положение 26 (рисунок 3.43) идентично положению 11 (см. рисунок 3.34).



Рисунок 3.43 – Положение платформы 3 ниже платформ 1, 2, 4 при движении на

спуске



Положение 27 (рисунок 3.44) идентично положению 10 (см. рисунок 3.33).

Рисунок 3.44 – Положение платформы 3 выше платформ 1, 2, 4 при движении на спуске

Положение 28 (рисунок 3.45) идентично положению 14 (см. рисунок 3.36).



Рисунок 3.45 – Положение платформы 4 ниже платформ 1, 2, 3 при движении на спуске

Положение 29 (рисунок 3.46) идентично положению 12 (см. рисунок 3.35).



Рисунок 3.46 – Положение платформы 4 выше платформ 1, 2, 3 при движении на

Положение 30 (рисунок 3.47) идентично положениям 19-21 (см. рисунок 3.38).



Рисунок 3.47 – Положение платформ 1-2 ниже платформ 3-4 при движении на спуске

Положение 31 (рисунок 3.48) идентично положениям 16-18 (см. рисунок 3.37).



Рисунок 3.48 – Положение платформ 1-2 выше платформ 3-4 при движении на спуске

#### 3.2 Синтез алгоритма работы системы управления

Нечеткий логический регулятор внешнего каскада призван регулировать положение гидроцилиндров полуплатформ для стабилизации всего трала в горизонтальном положении.

На рисунке 3.49 представлена структурная схема гидравлической системы полуплатформы с нечетким логическим регулятором третьего каскада, формирующим уставки внутренних модулей за счет суммирования значений сигналов управления [106].



1 – подструктура модели гидравлической станции; 2 – подструктуры тралового модуля

Рисунок 3.49 – Интеллектуальная система управления гидравлической подвеской полуплатформ с нечетким логическим регулятором, формирующим уставки внутренних модулей за счет суммирования значений сигналов управления

На рисунках 3.50 – 3.55 представлена настройка нечеткого логического регулятора с алгоритмом вывода Такаги-Сугено. Входные лингвистические переменные «sens\_plato\_1-2», «sens\_plato\_2-3» и «sens\_plato\_3-4» состоят из трех элементов в базовом терм-множестве и имеют треугольную функцию принадлежности, определяющие отклонение одной транспортной платформы относительно соседней транспортной платформы. Входные лингвистические переменные «Platform4\_Cilindr4» и «Platform1\_Cilindr1» состоят из семи элементов в базовом терммножестве и имеют треугольную функцию принадлежности, определяющие положение цилиндров 44 и 11 гидросистемы полуплатформ соответственно (рисунок 3.26). Выходные переменные представляют константы – сигнал управления на модули внутреннего каскада.



Рисунок 3.50 – Нечеткий модуль управления, формирующий уставки внутренних модулей за счет суммирования значений сигналов управления



Рисунок 3.51 – Входная лингвистическая переменная «sens\_plato\_1-2»



Рисунок 3.52 – Входная лингвистическая переменная «sens\_plato\_2-3»

98



Рисунок 3.53 – Входная лингвистическая переменная «sens\_plato\_3-4»



Рисунок 3.54 – Входная лингвистическая переменная «Platform4\_Cilindr4»

99



Рисунок 3.55 – Входная лингвистическая переменная «Platform1\_Cilindr1»

Нечеткая продукционная база знаний регулятора представляет собой перечень из 29 правил и имеет следующий вид:

1 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is null) and (sens\_plato\_3-4 is null) and (Platform4\_Cilindr4 is null) and (Platform1\_Cilindr1 is null) then (platform\_1 = null) (platform\_2 = null) (platform\_3 = null) (platform\_4 = null).

2 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is null) and (sens\_plato\_3-4 is null) and (Platform4\_Cilindr4 is high\_stop) and (Platform1\_Cilindr1 is low\_stop) then (platform\_1 = null) (platform\_2 = null) (platform\_3 = null) (platform\_4 = null).

3 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is null) and (sens\_plato\_3-4 is null) and (Platform4\_Cilindr4 is low\_stop) and (Platform1\_Cilindr1 is high\_stop) then (platform\_1 = null) (platform\_2 = null) (platform\_3 = null) (platform\_4 = null).

4 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is null) and (sens\_plato\_3-4 is null) and (Platform4\_Cilindr4 is low\_stop) then (platform\_1 = null) (platform\_2 = null) (platform\_4 = null).

5 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is null) and (sens\_plato\_3-4 is null) and (Platform1\_Cilindr1 is low\_stop) then (platform\_1 = null) (platform\_2 = null) (platform\_4 = null).

6 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is null) and (sens\_plato\_3-4 is null) and (Platform4\_Cilindr4 is null) and (Platform1\_Cilindr1 is low\_null) then (platform\_1 = null) (platform\_2 = null) (platform\_3 = null) (platform\_4 = null).

7 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is null) and (sens\_plato\_3-4 is null) and (Platform4\_Cilindr4 is null) and (Platform1\_Cilindr1 is high\_null) then (platform\_1 = null) (platform\_2 = null) (platform\_3 = null) (platform\_4 = null).

8 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is null) and (sens\_plato\_3-4 is null) and (Platform4\_Cilindr4 is low\_null) and (Platform1\_Cilindr1 is null) then (platform\_1 = null) (platform\_2 = null) (platform\_3 = null) (platform\_4 = null).

9 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is null) and (sens\_plato\_3-4 is null) and (Platform4\_Cilindr4 is high\_null) and (Platform1\_Cilindr1 is null) then (platform\_1 = null) (platform\_2 = null) (platform\_3 = null) (platform\_4 = null).

10 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is null) and (sens\_plato\_3-4 is null) and (Platform4\_Cilindr4 is high\_null) and (Platform1\_Cilindr1 is high\_null) then (platform\_1 = middle-low) (platform\_2 = middle-low) (platform\_3 = middle-low) (platform\_4 = middle-low).

11 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is null) and (sens\_plato\_3-4 is null) and (Platform4\_Cilindr4 is low\_null) and (Platform1\_Cilindr1 is low\_null) then (platform\_1 = middle-high) (platform\_2 = middle-high) (platform\_3 = middle-high) (platform\_4=middle-high).

12 If (sens\_plato\_1-2 is low) and (sens\_plato\_2-3 is null) and (sens\_plato\_3-4 is null) and (Platform1\_Cilindr1 not is high\_stop) then (platform\_1 = middle-high) (platform\_2 = null) (platform\_3 = null) (platform\_4= null).

13 If (sens\_plato\_1-2 is low) and (sens\_plato\_2-3 is null) and (sens\_plato\_3-4 is null) and (Platform1\_Cilindr1 is high\_stop) then (platform\_1 = null) (platform\_2 = middle-low) (platform\_3 = middle-low) (platform\_4= middle-low).

14 If (sens\_plato\_1-2 is high) and (sens\_plato\_2-3 is null) and (sens\_plato\_3-4 is null) and (Platform1\_Cilindr1 not is low\_stop) then (platform\_1 = middle-low) (platform\_2 = null) (platform\_3 = null) (platform\_4= null).

15 If (sens\_plato\_1-2 is high) and (sens\_plato\_2-3 is null) and (sens\_plato\_3-4 is null) and (Platform1\_Cilindr1 is low\_stop) then (platform\_1 = null) (platform\_2 = middle-high) (platform\_3 = middle-high) (platform\_4= middle-high).

16 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is null) and (sens\_plato\_3-4 is low) and (Platform4\_Cilindr4 not is low\_stop) then (platform\_1 = null) (platform\_2 = null) (platform\_3 = null) (platform\_4 = middle-low).

17 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is null) and (sens\_plato\_3-4 is low) and (Platform4\_Cilindr4 is low\_stop) then (platform\_1 = middle-high) (platform\_2 = middle-high) (platform\_3 = middle-high) (platform\_4= null).

18 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is null) and (sens\_plato\_3-4 is high) and (Platform4\_Cilindr4 not is high\_stop) then (platform\_1 = null) (platform\_2 = null) (platform\_3 = null) (platform\_4 = middle-high).

19 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is null) and (sens\_plato\_3-4 is high) and (Platform4\_Cilindr4 is high\_stop) then (platform\_1 = middle-low) (platform\_2 = middle-low) (platform\_3 = middle-low) (platform\_4 = null).

20 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is low) and (sens\_plato\_3-4 is high) then (platform\_1 = null) (platform\_2 = null) (platform\_3 = middle-low) (platform\_4= null).

21 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is high) and (sens\_plato\_3-4 is low) then (platform\_1 = null) (platform\_2 = null) (platform\_3 = middle-high) (platform\_4= null).

22 If (sens\_plato\_1-2 is low) and (sens\_plato\_2-3 is high) and (sens\_plato\_3-4 is null) then (platform\_1 = null) (platform\_2 = middle-low) (platform\_3 = null) (platform\_4= null).

23 If (sens\_plato\_1-2 is high) and (sens\_plato\_2-3 is low) and (sens\_plato\_3-4 is null) then (platform\_1 = null) (platform\_2 = middle-high) (platform\_3 = null) (platform\_4= null).

24 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is low) and (sens\_plato\_3-4 is null) and (Platform4\_Cilindr4 not is low\_stop) and (Platform1\_Cilindr1 not is

high\_stop) then (platform\_1 = middle-high) (platform\_2 = middle-high) (platform\_3 = middle-low) (platform\_4= middle-low).

25 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is low) and (sens\_plato\_3-4 is null) and (Platform4\_Cilindr4 not is low\_stop) and (Platform1\_Cilindr1 is high\_stop) then (platform\_1 = null) (platform\_2 = null) (platform\_3 = middle-low) (platform\_4= middle-low).

26 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is low) and (sens\_plato\_3-4 is null) and (Platform4\_Cilindr4 is low\_stop) and (Platform1\_Cilindr1 not is high\_stop) then (platform\_1 = middle-low) (platform\_2 = middle-low) (platform\_3 = null) (platform\_4= null).

27 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is high) and (sens\_plato\_3-4 is null) and (Platform4\_Cilindr4 not is high\_stop) and (Platform1\_Cilindr1 not is low\_stop) then (platform\_1 = middle-low) (platform\_2 = middle-low) (platform\_3 = middle-high) (platform\_4= middle-high).

28 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is high) and (sens\_plato\_3-4 is null) and (Platform4\_Cilindr4 is high\_stop) and (Platform1\_Cilindr1 not is low\_stop) then (platform\_1 = middle-low) (platform\_2 = middle-low) (platform\_3 = null) (platform\_4= null).

29 If (sens\_plato\_1-2 is null) and (sens\_plato\_2-3 is high) and (sens\_plato\_3-4 is null) and (Platform4\_Cilindr4 not is high\_stop) and (Platform1\_Cilindr1 is low\_stop) then (platform\_1 = null) (platform\_2 = null) (platform\_3 = middle-high) (platform\_4= middle-high).

Правила 1-11 используются нечеткой системой управления для проверки всего трала на горизонтальное положение, выполняя функцию синхронизации.

Правила 12-29 позволяют нечеткой системой управления определять отклонения трала от горизонтального положения для дальнейшего формирования управляющего воздействия.

На рисунке 3.56 представлены графики входных параметров, поступающих в нечеткий логический регулятор.

На рисунке 3.57 представлены графики переходных процессов в гидроприводах с использованием нечеткого модуля управления внешнего каскада, формирующего уставки внутренних модулей за счет суммирования значений сигналов управления. Данный регулятор за счет сигналов, полученных от датчиков положения платформ (энкодеров) и датчиков положения штоков гидроцилиндров, позволяет регулировать положение полуплатформ относительно друг друга, что способствует удержанию всего трала в одной плоскости.

На рисунках 3.58 – 3.60 представлены графики переходных процессов при подаче на систему возмущающих воздействий в виде белого шума, синусоидального сигнала с белым шумом, что позволят смоделировать неровность дорожного покрытия, а также локальные экстремумы [84].



Рисунок 3.56 – Графики переходных процессов входных параметров, поступающих в нечеткий логический регулятор



Рисунок 3.57 – Графики переходного процесса в гидроприводах с использованием нечеткого модуля управления внешнего каскада, формирующего уставки внутренних модулей за счет суммирования значений сигналов управления



Рисунок 3.58 – Графики переходных процессов в гидроприводах при подаче на систему возмущающего воздействия

в виде белого шума

#### в виде синусоидального сигнала




Рисунок 3.60 – Графики переходных процессов в гидроприводах при подаче на систему возмущающего воздействия в виде синусоидального сигнала с белым шумом

В интервале времени с 1 до 6,75 с (зона I) на вход к НЛР поступают сигналы с датчиков положения гидроцилиндров 11 и 44 (см. рисунок 3.56) об одновременном втягивании этих гидроцилиндров, при этом сигналы с датчиков 1-2, 2-3 и 3-4 равны нулю, что соответствует отсутствию наклона между полуплатформами. Реагируя на входные значения, НЛР формирует управляющий сигнал в соответствии с правилом 11 – выдвинуть все гидроцилиндры четырех полуплатформ, позволяющий скорректировать положение трала.

Через некоторое время сигналы с датчиков положения гидроцилиндров 11 и 44 меняются на противоположные, что означает о выдвижении гидроцилиндров. Реагируя на это, НЛР формирует управляющий сигнал в соответствии с правилом 10 – втянуть все гидроцилиндры четырех полуплатформ, позволяющий скорректировать положение трала.

В интервале времени с 8 до 13,75 с (зона II) на вход к НЛР поступает сигнал с датчика 2-3 положения полуплатформ 2 и 3 (см. рисунок 3.56), при этом сигналы с датчиков 1-2 и 3-4 равны нулю. Гидроцилиндры 11 и 44 находятся в среднем положении. Данный сигнал соответствует положению 19 (см. рисунок 3.38). Реагируя на входные значения, НЛР формирует управляющий сигнал в соответствии с правилом 24 – выдвинуть все штоки полуплатформ 1 и 2 и втянуть все штоки полуплатформ 3 и 4, позволяющий скорректировать положение трала. Через некоторое время сигнал с датчика 2-3 положения полуплатформ 2 и 3 меняется на противоположный, что соответствует положению 16 (см. рисунок 3.48). Реагируя на входные значения, НЛР формирует управляющий сигнал в соответствии с правилом 27 – втянуть все штоки полуплатформ 1 и 2 и выдвинуть все штоки полуплатформ 3 и 4, позволяющий скорректировать положение трала.

В интервале времени с 14,5 до 19 с (зона III) на вход к НЛР поступает сигнал с датчика 1-2 положения полуплатформ 1 и 2 (см. рисунок 3.56), при этом сигналы с датчиков 2-3 и 3-4 равны нулю. Гидроцилиндры 11 и 44 находятся в среднем положении. Данный сигнал соответствует положению 4 (см. рисунок 3.28). Реагируя на входные значения, НЛР формирует управляющий сигнал в соответствии с правилом 14 – втянуть все штоки полуплатформы 1, не изменять по-

110

ложение гидроцилиндров полуплатформ 2, 3 и 4, позволяющий скорректировать положение трала.

В интревале времени с 19,5 до 24 с (зона IV) на вход к НЛР поступает сигнал с датчика 1-2 положения полуплатформ 1 и 2 (см. рисунок 3.56), при этом сигналы с датчиков 2-3 и 3-4 равны нулю. Гидроцилиндр 44 находится в среднем положении, гидроцилиндр 11 выдвинут практически на упор. Данный сигнал соответствует положению 5 (см. рисунок 3.28), но так как гидроцилиндр 11 выдвигаться более не может, то использование правила 10 невозможно. Реагируя на входные значения, НЛР формирует управляющий сигнал в соответствии с правилом 15 – все штоки полуплатформ 2, 3 и 4 втянуть, не изменять положение гидроцилиндров полуплатформы 1, позволяющий скорректировать положение трала.

## Выводы по третьей главе

Разработан алгоритм функционирования системы управления процессом стабилизации груза на комбинированном траловом механизме для перемещения высокотоннажных крупногабаритных грузов, отличительной особенностью которой является формирование управляющего воздействия многокаскадным нечетким регулятором на основании контроля входных параметров. Сформирована база правил для нечеткого регулятора внутреннего каскада, регулирующего положение гидроцилиндра, а также внешнего каскада, обеспечивающего общую стабилизацию всех платформ по положению.

Анализируя модели предложенных многокаскадных регуляторов с различными операторами при сопряжении каскадов, можно сделать вывод, что применение функции суммирования позволяет реализовать необходимые законы управления с более высокой точностью, адаптивностью к ряду внешних возмущающих факторов, а также с упрощенной понятийной составляющей при настройке интеллектуальной системы. Кроме того, можно отметить, что качество настройки развитой интеллектуальной системы управления в меньшей степени зависит от выбора сочетания механизмов выводов и скорее продиктовано особенностями функционирования самого объекта регулирования.

# Глава 4 Практическая реализация нечетких алгоритмов управления с использованием промышленных средств автоматизации

Реализация нечетких логических регуляторов на аппаратных платформах имеет свои особенности, связанные как с вычислительной сложностью алгоритмов, так и с требованиями к быстродействию и энергопотреблению. Одной из ключевых задач является эффективное выполнение операций нечеткого вывода, включая фаззификацию (преобразование входных данных в нечеткие множества), применение правил нечеткой логики и дефаззификацию (преобразование нечетких выводов в четкие значения). На аппаратных платформах, таких как микроконтроллеры (например, STM32, Arduino) или программируемые логические интегральные схемы (ПЛИС, FPGA), важно оптимизировать вычисления для минимизации задержек и использования ресурсов. Это может включать применение упрощенных методов дефаззификации (например, метода центра тяжести) или использование предварительно рассчитанных таблиц для ускорения обработки правил [80, 82].

Еще одной особенностью реализации НЛР на аппаратных платформах является необходимость учета ограничений по памяти и вычислительной мощности. Микроконтроллеры с ограниченными ресурсами требуют тщательной оптимизации кода, включая использование целочисленной арифметики вместо операций с плавающей точкой, что может снизить точность, но значительно ускорить выполнение. Ключевым аспектом применения микроконтроллеров STM32 для реализации многокаскадного нечеткого регулятора является наличие DSP-инструкций (Digital Signal Processing), доступных в ядрах Cortex-M4 и Cortex-M7, позволяюцих оптимизировать обработку данных за счет параллелизма вычислений [87, 90, 111].

Многокаскадный нечеткий логический регулятор, отвечающий за управление одной четвертой части модульного тралового механизма платформы (см. рисунок 3.10), состоящий из каскада первого и второго уровней вложенности, реализован на базе программируемого микроконтроллера STM32F746ZGT6, ARM Cortex-M7 (рисунок 4.1). Синтезированный сигнал управления многокаскадного нечеткого логического регулятора на микроконтроллере передается на имитационную модель.



Рисунок 4.1 – Экспериментальный стенд для проверки многокаскадного нечеткого логического регулятора первого и второго уровней вложенности

Разработанный в главе 3 многокаскаднй нечеткий логический регулятор, особенностью которого является синтез уставок для регулятора вложенного каскада второго уровня на основе оператора суммы, с помощью специальных средств был экспортирован в микроконтроллер STM32. Используя режим Virtual COM Port, который позволяет микроконтроллеру эмулировать последовательный порт (UART) через USB, данные отклонений по осям абцисс и ординат, а также положение гидроприводов поступают в микроконтроллер в реальном масштабе времени. На основании полученных данных на микроконтроллере синтезируется сигнал управления и обратно передается на имитационную модель одного модуля тралового механизма, развернутую на персональном компьютере. Полученные графики положения приводов с управляющими сигналами от микроконтроллера сравниваются с сигналами эталонной имиитационной модели одного модуля тралового механизма (рисунок 4.2).



Рисунок 4.2 – Функциональная схема взаимодействия контроллера STM32 с имитационной моделью

Графики переходных процессов в гидроприводах по положению представлены на рисунке 4.3.

Регулятор стабилизирует положение платформы при возникновении продольных и поперечных возмущений в следующих временных периодах (см. рисунок 4.3): время с 5 до 10 с – отклонение  $X = 0^\circ$ ,  $Y = +9^\circ$ ; время с 15 до 20 с – отклонение  $X = -6^\circ$ ,  $Y = 0^\circ$ ; время с 25 до 30 с – отклонение  $X = +3^\circ$ ,  $Y = -4,5^\circ$ ; время с 35 до 40 с – отклонение  $X = -5^\circ$ ,  $Y = -6^\circ$ .

Из анализа полученных графиков переходных процессов можно сделать вывод, что результаты моделирования системы управления с многокаскадным нечетким логическим регулятором, реализуемые микроконтроллером, совпадают с графиками эталонной имитационной модели. Некоторое отклонение при отработке основных параметров переходных процессов многокаскадной нечеткой системой управления, реализованной на базе программируемого логического контроллера, обусловлено частотой дискретизации сигналов [68].



 эталонная имитационная модель; 2 – аппаратная реализация многокаскадного нечеткого регулятора

Рисунок 4.3 – Графики переходных процессов по положению гидроприводов

## Выводы по четвертой главе

Процедуры нечеткого многокаскадного управления, сформированные на платформе программируемого логического контроллера, показали достаточно высокое качество при реализации законов регулирования модулем комбинированного тралового механизма. Отклонение интеллектуальной системы управления от эталонной модели показало достаточно малые значения и составило не более 1,2 %.

Для апробации сформированных законов управления разработана автоматизированная экспериментальная установка процесса стабилизации груза с интеллектуальной системой управления на основе многокаскадного нечеткого логического регулятора. Объединение гидроприводов с электронными системами управления и применением интеллектуальной системы управления позволяет успешно сочетать силовые и динамические качества гидравлики с высокими показателями качества переходных процессов регуляторов, построенных на базе нечеткой логики.

## Заключение

В диссертационной работе решена научно-техническая задача по созданию многокаскадной нечеткой системы, реализующей сложные законы управления для стабилизации модульного тралового механизма при перемещении высокотоннажных грузов с учетом действия недетерминированных возмущений. Разработаны структурные и параметрические решения для моделей и алгоритмов, реализующих процедуры управления на основе многокаскадного нечеткого логического регулятора.

По результатам диссертационного исследования сделаны следующие выводы:

1 На основании проведенного анализа существующих моделей и алгоритмов управления составной многокоординатной системой на примере модульного тралового механизма определен ряд параметров, действующих на объект и оказывающих существенное влияние на его положение, учет которых при стандартных способах регулирования невозможен либо затруднен.

2 Предложено математическое описание нечеткого логического регулятора с многокаскадной структурой построения, особенностью которой является формирование уставок для элементов вложенного каскада на основе операторов произведения и суммы, позволяющего повысить возможности применения модульного тралового механизма на дорогах общего пользования, а также обеспечить независимость системы регулирования от компоновки его составных частей.

3 Разработана имитационная модель многокаскадной нечеткой системы управления, которая позволила выявить влияние продольных и поперечных кренов составляющих полуплатформ, а также сформировать интеллектуальную базу знаний регуляторов внешнего и внутреннего каскада первого и второго уровней вложенности, которая обеспечивает стабилизацию всех элементов модульного тралового механизма при различной компоновке и невыход продольных и поперечных отклонений каждой полуплатформы за пределы допустимых значений  $[\pm 6^\circ; \pm 9^\circ].$  4 Показано, что применение иерархической многокаскадной структуры нечеткого логического регулятора позволяет сократить информационную нагрузку на средства промышленной автоматики, упростить реализацию интеллектуальной системы управления локального уровня и интегрировать в нее в существующие системы регулирования элементами подвесной системы модульного тралового механизма на имитационной модели.

Перспективы дальнейшего развития работы. Предполагается развитие полученных результатов исследования с применением многокаскадного нечеткого логического регулятора с большим числом внутренних модулей для многодвигательных систем, штыревых машин, беспилотных аппаратов.

#### Список использованных источников

1 Аксенов, П. В. Многоосные автомобили / П. В. Аксенов. – 2-е издание, переработаное и дополненное. – М. : Машиностроение, 1989. – 280 с.

2 Алиев, Р. А. Управление производством при нечеткой исходной информации / Р. А. Алиев, А. Э. Церковный, Г. А. Мамедова. – М. : Энерго-атомиздат, 1991. – 240 с.

3 Андреев, С. М. Разработка и моделирование несложных систем автоматизации с учетом специфики технологических процессов : учебное пособие / С. М. Андреев, Б. Н. Парсункин. – М. : Академия, 2016. – 272 с.

4 Башта, Т. М. Машиностроительная гидравлика / Т. М. Башта. – М. : Машиностроение, 1971. – 152 с.

5 Белов, М. П. Технические средства автоматизации и управления / М. П. Белов. – СПб. : СЗТУ, 2006. – 184 с.

6 Бесекерский, В. А. Теория систем автоматического управления : учебное пособие / В. А. Бесекерский, Е. П. Попов. – СПб. : Профессия, 2003. – 752 с.

7 Блишун, А. Ф. Обоснование операций теории нечетких множеств. Нетрадиционные модели и системы с нечеткими знаниями / А. Ф. Блишун, С. Ю. Знатнов. – М. : Энергоатомиздат, 1991. – С. 21–33.

8 Борисевич, А. В. Теория автоматического управления : элементарное введение с применением MATLAB [Электронный ресурс] / А. В. Борисевич. – М. : ИнфраМ, 2014. – 200 с.

9 Брадис, В. М. Четырехзначные математические таблицы / В. М. Брадис. – М. : Дрофа, 2021. – 96 с.

10 Бронштейн, И. Н. Справочник по математике для инженеров и учащихся вузов / И. Н. Бронштейн, К. А. Семендяев. – М. : Наука, 1986. – 544 с.

11 Буланов, С. В. Обоснование состава, структуры и параметров индивидуального рулевого привода с микропроцессорным управлением и привода ведущих колес транспортных средств большой грузоподъемности с применением модульного принципа создания / С. В. Буланов // Автомобиль. Дорога. Инфраструктура.  $-2015. - N_{2} 1. - C. 1-8.$ 

12 Верещагин, С. Б. Направления развития многоопорных самоходных большегрузных транспортных средств / С. Б. Верещагин, Г. И. Гладов, Л. В. Демидов // Автомобильная промышленность. – 2014. – № 9. – С. 7–9.

13 Воронов, А. А. Основы теории автоматического управления Ч. 3 / А. А. Воронов. – Ленинград : Энергия, 1970. – 328 с.

14 Востриков, А. С. Синтез систем регулирования методом локализации : монография / А. С. Востриков. – Новосибирск : НГТУ, 2007. – 252 с.

15 Востриков, А. С. Теория автоматического регулирования : учебное пособие / А. С. Востриков, Г. А. Французова. – Новосибирск : НГТУ, 2006. – 368 с.

16 Галдин, Н. С. Атлас гидравлических схем мобильных машин и оборудования : учебное пособие / Н. С. Галдин, А. В. Кукин. – Омск : СибАДИ, 2006. – 91 с.

17 Гидравлика и гидропривод : практикум / В. Е. Щерба, Е. А. Павлюченко, Е. Ю. Носов [и др.]. – Омск : ОмГТУ, 2020. – 187 с.

18 Гидравлика / составители : В. В. Крамаренко, О. Г. Савичев. – Томск : ТПУ, 2009. – 124 с.

19 Гидравлика, гидромашины и гидроприводы : учебник для машиностроительных вузов / Т. М. Башта, С. С. Руднев, Б. Б. Некрасов [и др.]. – 2-е издание, переработанное. – М. : Машиностроение, 1982. – 423 с.

20 Гидравлические и Пневматические системы транспортных и транспортно-технологических машин и оборудования : учебное пособие / В. В. Буренин [и др.]. – М. : ФГУП «ЦЭНКИ»-КБТХМ, 2017. – 214 с.

21 Гидравлический расчет объемного гидропривода с возвратнопоступательным движением выходного звена : методические указания / составитель Н. П. Жуков. – Тамбов : ГОУ ВПО ТГТУ, 2010. – 32 с.

22 Гладов, Г. И. Типы многоопорных самоходных большегрузных транспортных средств и особенности их движения по криволинейной траектории / Г. И. Гладов, Л. В. Демидов // Вестник МАДИ. – 2014. – Т. 37, вып. 2. – С. 43–47. 23 Глазырин, Г. В. Теория автоматического регулирования / Г. В. Глазырин. – Новосибирск : НГТУ, 2014. – 168 с.

24 Горюшкин, В. А. Математические модели с логическими регуляторами / В. А. Горюшкин // Вестник Камчатского государственного технического университета. – 2012. – С. 5–14.

25 Григорьева, Д. Р. Основы нечеткой логики : учебно-методическое пособие к практическим занятиям и лабораторным работам / Д. Р. Григорьева, Г. А. Гареева, Р. Р. Басыров. – Набережные Челны : НЧИ КФУ, 2018. – 42 с.

26 Гроховский, Д. В. Основы гидравлики и гидропривод : учебное пособие / Д. В. Гроховский. – СПб. : Политехника, 2013. – 236 с.

27 Деменков, Н. П. Программные средства оптимизации настройки систем управления / Н. П. Деменков. – М. : МГТУ имени Н. Э. Баумана, 2006. – 244 с.

28 Деменков, Н. П. Нечеткое управление в технических системах / Н. П. Деменков. – М. : МГТУ имени Н. Э. Баумана, 2005. – 200 с.

29 Демидова, Г. Л. Регуляторы на основе нечеткой логики в системах управления техническими объектами / Г. Л. Демидова, Д. В. Лукичев. – СПб. : Университет ИТМО, 2017. – 81 с.

30 Жмудь, В. А. Моделирование, исследование и оптимизация замкнутых систем автоматического управления / В. А. Жмудь. – Новосибирск : НГТУ, 2012. – 335 с.

31 Заде, Л. А. Основы нового подхода к анализу сложных систем и процессов принятия решений / Л. А. Заде. – М. : Знание, 1974. – 64 с.

32 Заде, Л. Понятие лингвистической переменной и его применение к принятию приближенных решений : монография / Л. Заде. – М. : Мир, 1976. – 166 с.

33 Захаров, В. Н. Искусственный интеллект. Программные и аппаратные средства : справочник / В. Н. Захаров, В. Ф. Хорошевский. – М. : Радио и связь, 1990. – 368 с.

34 Зудин, В. Л. Датчики: измерение перемещений, деформаций и усилий : учебное пособие для вузов / В. Л. Зудин, Ю. П. Жуков, А. Г. Маланов. – 2-е издание. – М. : Юрайт, 2022 ; Ярославль : ЯГТУ. – 199 с.

35 Иващенко, Н. Н. Автоматическое регулирование. Теория и элементы систем : учебник для вузов / Н. Н. Иващенко. – 4-е издание, переработанное и дополненное. – М. : Машиностроение, 1978. – 736 с.

36 Карпов, А. С. Анализ конструкции гидроцилиндров рабочего оборудования дорожных, строительных машин / А. С. Карпов, С. М. Давтян, В. В. Жмуров // Механики XXI веку. – 2019. – № 18. – С. 97–100. – EDN VBRRUU.

37 Кишкевич, П. Н. Статический и динамический расчет гидро- и пневмораспределителей : учебное пособие / П. Н. Кишкевич, М. И. Жилевич, П. Р. Бартош. – Минск : БНТУ, 2012. – 80 с.

38 Классификация транспортных средств : учебное пособие / А. В. Острецов,
Б. Н. Белоусов [и др.]. – М. : МГТУ «МАМИ», 2011. – 71 с.

39 Коваль, А. С. К вопросу построения нечеткого регулятора скорости для электропривода при прямом управлении моментом с широтно-импульсным модулятором лифтовой установки / А. С. Коваль, А. В. Шваяков // Вестник Белорусско – Российского университета. – 2008. – № 1. – С. 114–122.

40 Колесные транспортные средства особо большой грузоподъемности. Конструкция. Теория. Расчет / под общей редакцией Б. Н. Белоусова. – М. : МГТУ имени Н. Э. Баумана, 2006. – 728 с.

41 Конструкции многоцелевых гусеничных и колесных машин : учебник для студентов высших учебных заведений / Г. И. Гладов [и др.]. – М. : Академия, 2010. – 400 с.

42 Котович, С. В. Движители специальных транспортных средств : учебное пособие. Часть I / С. В. Котович. – М. : МАДИ (ГТУ), 2008. – 161 с.

43 Круглов, В. В. Нечеткая логика и искусственные нейронные сети / В. В. Круглов, М. И. Дли, Р. Ю. Голунов. – М. : ФИЗМАТЛИТ, 2001. – 201 с.

44 Кудинов, Ю. И. Нечеткие регуляторы и системы управления / Ю. И. Кудинов, И. Н. Дорохов, Ф. Ф. Пащенко // Проблемы управления. – 2004. – № 3. – С. 2–14.

45 Кудинов, Ю. И. Нечеткие системы управления / Ю. И. Кудинов // Техническая кибернетика. – 1990. – № 5. – С. 196–206. 46 Леоненков, А. В. Нечеткое моделирование в среде MATLAB и fuzzyTECH / А. В. Леоненков. – СПб. : БХВ-Петербург, 2005. – 736 с.

47 Ловкис, З. В. Гидравлика : учебное пособие / З. В. Ловкис, Б. А. Карташов, П. В. Лаврухин. – Ростов-на-Дону : Феникс, 2019. – 384 с.

48 Малиновский, М. П. Системы управления колесных машин : учебное пособие / М. П. Малиновский. – М. : МАДИ, 2018. – 100 с.

49 Мелихов, А. Н. Ситуационные советующие системы с нечеткой логикой /
А. Н. Мелихов, Л. С. Берштейн, С. Я. Коровин. – М. : Наука, 1990. – 272 с.

50 Модели и алгоритмы для интеллектуальных систем управления / А. А. Богуславский, Г. К. Боровин, В. А. Карташев [и др.]. – М. : ИПМ имени М. В. Келдыша, 2019. – 228 с.

51 Нечеткие множества в моделях управления и искусственного интеллекта / А. Н. Аверкин, И. З. Батыршин, А. Ф. Блишун [и др.] / под редакцией Д. А. Поспелова. – М. : Наука, 1986. – 312 с.

52 Опейко, А. В. Моделирование и синтез управляемой подвески многоопорных машин : специальность 05.05.03 : автореферат диссертации на соискание ученой степени доктора технических наук / Опейко Александр Федорович. – Минск, 1993. – 35 с.

53 Орлов, А. И. Задачи оптимизации и нечеткие переменные / А. И. Орлов. – М. : Знание, 1980. – 64 с.

54 Павлов, В. В. Теория движения многоцелевых гусеничных и колесных машин : учебник для вузов / В. В. Павлов, В. В. Кувшинов. – Чебоксары : ООО «Чебоксарская типография № 1», 2011. – 424 с.

55 Панкратов, В. В. Специальные разделы современной теории автоматического управления : учебное пособие / В. В. Панкратов, Е. А. Зима, О. В. Нос. – Новосибирск : НГТУ, 2007. – 220 с.

56 Парсункин, Б. Н. Локальные стабилизирующие контуры автоматического управления в АСУ ТП промышленного производства : монография / Б. Н. Парсункин, С. М. Андреев, О. С. Логунова. – Магнитогорск : КТ «Буки-Веди», 2012. – 406 с.

57 Пегат, А. Нечеткое моделирование и управление / А. Пегат ; перевод с английского А. Г. Подвесовского ; под редакцией Ю. В. Тюменцева. – М. : БИНОМ. Лаборатория знаний, 2013. – 798 с.

58 Петрова, А. М. Автоматическое управление : учебное пособие / А. М. Петрова. – М. : Форум, 2010. – 240 с.

59 Поспелов, Д. А. Логико-лингвистические методы в системах управления /Д. А. Поспелов. – М. : Энергоатомиздат, 1981. – 190 с.

60 Почебут, М. В. Адаптивно-робастная дискретная система фазового управления : специальность 05.13.01 : автореферат диссертации на соискание ученой степени кандидата технических наук / М. В. Почебут ; научный руководитель А. А. Лобатый. – Минск : БГУИР, 2011. – 22 с.

61 Проектирование полноприводных колесных машин : учебник для вузов : в 2-х т. Т. 1 / Б. А. Афанасьев, Н. Ф. Бочаров, Л. Ф. Жеглов [и др.] / под общей редакцией А. А. Полунгяна. – М. : МГТУ имени Н. Э. Баумана, 2008. – 496 с.

62 Проектирование полноприводных колесных машин : учебник для вузов : в 2-х т. Т. 2 / Б. А. Афанасьев, Н. Ф. Бочаров, Л. Ф. Жеглов [и др.] / под общей редакцией А. А. Полунгяна. – М.: МГТУ имени Н. Э. Баумана, 2009. – 640 с.

63 Рассел, С. Искусственный интеллект: современный подход : научное пособие / С. Рассел, П. Норвиг. – М. : Вильяме, 2010. – 408 с.

64 Рогов, В. А. Технические средства автоматизации и управления : учебник для среднего профессионального образования / В. А. Рогов, А. Д. Чудаков. – 2-е издание, исправленное и дополненное. – М. : Юрайт, 2023. – 352 с.

65 Ротенберг, Р. В. Подвеска автомобиля / Р. В. Ротенберг. – М. : Машиностроение, 1972. – 392 с.

66 Рульнов, А. А. Автоматическое регулирование : учебник / А. А. Рульнов, И. И. Горюнов, К. Ю. Евстафьев. – 2-е издание, стереотипное. – М. : НИЦ Инфра-М, 2013. – 219 с.

67 Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы : монография/ Д. Рутковская, М. Пилинский, Л. Рутковский. – М. : Горячая линия – Телеком, 2004. – С. 452. 68 Рябчиков, М. Ю. Планирование эксперимента и обработка результатов измерений : учебное пособие / М. Ю. Рябчиков, Е. С. Рябчикова. – Магнитогорск : ГОУ ВПО «МГТУ имени Г. И. Носова», 2012. – 150 с.

69 Сафиуллин, Р. К. Основы автоматики и автоматизация процессов : учебное пособие для вузов / Р. К. Сафиуллин. – 2-е издание, исправленное и дополненное. – М. : Юрайт, 2023. – 146 с.

70 Свидетельство о государственной регистрации программы для ЭВМ № 2023617229 Российская Федерация. Программа реализации нечеткого регулятора для системы управления маломощным электроприводом : № 2023616075 ; заявл. 28.03.2023 ; опубл. 06.04.2023 / В. Н. Хрульков, С. П. Черный, А. И. Бузикаева ; заявитель и правообладатель ФГБОУ ВО «КнАГУ». – 1 с.

71 Свидетельство о государственной регистрации программы для ЭВМ № 2023617058 Российская Федерация. Программная реализация нечеткого логического регулятора для управления модульной платформой по перемещению крупногабаритных высокотонажных объектов : № 2023615963 ; заявл. 28.03.2023 ; опубл. 05.04.2023 / В. Н. Хрульков, С. П. Черный, А. И. Бузикаева ; заявитель и правообладатель ФГБОУ ВО «КнАГУ». – 1 с.

72 Свидетельство о государственной регистрации программы для ЭВМ № 2024669082 Российская Федерация. Многокаскадная нечеткая система управления модульным тралом со стабилизацией горизонтального положения : № 2024668216 ; заявл. 31.07.2024 ; опубл. 14.08.2024 / В. Н. Хрульков, С. П. Черный, А. В. Охотников ; заявитель и правообладатель ФГБОУ ВО «КнАГУ». – 1 с.

73 Серебряков, А. С. Автоматика : учебник и практикум для среднего профессионального образования / А. С. Серебряков, Д. А. Семенов, Е. А. Чернов ; под общей редакцией А. С. Серебрякова. – 2-е издание, исправленное и дополненное. – М. : Юрайт, 2023. – 476 с.

74 Системы управления технологическими процессами и информационные технологии / В. В. Троценко [и др.]. – М. : Юрайт, 2017. – 160 с.

75 Соловьев, В. А. Искусственный интелект в задачах управления. Интеллектуальные системы управления технологическими процессами : учебное пособие / В. А. Соловьев, С. П. Черный. – Владивосток : Дальнаука, 2010. – 267 с.

76 Сологуб, В. А. Автопрактикум : методические указания : в 3-х ч. Ч 2: Трансмиссия большегрузных автомобилей / В. А. Сологуб, И. В. Сологуб. – Оренбург : ОГУ, 2012. – 33 с.

77 Теория и проектирование гидропневмоприводов : конспект лекций : в 6 ч.
Ч. 2: Следящие гидро- и пневмоприводы с дроссельным управлением / В. П. Автушко [и др.]. – Минск : БНТУ, 2017. – 122 с.

78 Трифонова, Г. О. Гидропневмопривод: следящие системы приводов : учебное пособие для вузов / Г. О. Трифонова, О. И. Трифонова. – 2-е издание, исправленное и дополненное. – М. : Юрайт, 2021. – 140 с.

79 Хрульков, В. Н. Моделирование многокаскадного нечеткого логического регулятора для управления многосоставным траловым механизмом на основе оператора сложения / В. Н. Хрульков, С. П. Черный // Ученые записки Комсомольского-на-Амуре государственного технического университета. Науки о природе и технике. – 2023. – № 5(69). – С. 53–60.

80 Хрульков, В. Н. Особенности реализации нечетких алгоритмов управления на базе программируемых логических контроллеров / В. Н. Хрульков, С. П. Черный // Ученые записки Комсомольского-на-Амуре государственного технического университета. Науки о природе и технике. – 2022. – № 1(57). – С. 52–62.

81 Хрульков, В. Н. Моделирование элементов гидравлической системы трала / В. Н. Хрульков, С. П. Черный // Молодежь и наука: актуальные проблемы фундаментальных и прикладных исследований : материалы IV Всероссийской национальной научной конференции студентов, аспирантов и молодых ученых : в 4-х частях, Комсомольск-на-Амуре, 12 – 16 апреля 2021 года. Часть 2. – Комсомольск-на-Амуре : ФГБОУ ВО «КнАГУ», 2021. – С. 83–85.

82 Хрульков, В. Н. Анализ нечетких систем реализованных на ПЛК и средствами моделирования / В. Н. Хрульков, С. П. Черный, Т. А. Давудов // Молодежь и наука: актуальные проблемы фундаментальных и прикладных исследований : материалы III Всероссийской национальной научной конференции студентов, аспирантов и молодых ученых : в 3-х частях, Комсомольск-на-Амуре, 06 – 10 апреля 2020 года. Часть 1. – Комсомольск-на-Амуре : ФГБОУ ВО «КнАГУ», 2020. – С. 352–354.

83 Хрульков, В. Н. К вопросу моделирования нечетких систем управления гидравлической подвески трала / В. Н. Хрульков, С. П. Черный // XLVII Гагаринские чтения 2021 : сборник тезисов работ XLVII международной молодежной научной конференции, Москва, 20 – 23 апреля 2021 года. – Москва : Перо, 2021. – С. 756–757.

84 Хрульков, В. Н. Моделирование многокаскадного нечеткого логического регулятора для стабилизации модульного тралового механизма / В. Н. Хрульков, С. П. Черный, Б. Я. Мокрицкий // Современные наукоемкие технологии. – 2024. – № 11. – С. 126–132. – DOI 10.17513/snt.40218. – EDN KMGSTW.

85 Хрульков, В. Н. Один из подходов к моделированию процесса стабилизации высокотоннажной платформы / В. Н. Хрульков, С. П. Черный, А. В. Охотников // Производственные технологии будущего: от создания к внедрению : материалы V Международной научно-практической конференции, Комсомольск-на-Амуре, 06 – 11 декабря 2021 года / редколлегия : С. И. Сухоруков (отв. ред.), А. С. Гудим, Н. Н. Любушкина. – Комсомольск-на-Амуре : ФГБОУ ВО «КнАГУ», 2022. – С. 100–103.

86 Хрульков, В. Н. Особенности реализации интеллектуальной развитой системы управления модульным высокотоннажным тралом / В. Н. Хрульков, А. В. Бузикаева, С. П. Черный // Трансформация информационно-коммуникативной среды общества в условиях вызовов современности : материалы II Международной научно-практической конференции молодых ученых, Комсомольск-на-Амуре, 30 ноября – 01 2023 года. – Комсомольск-на-Амуре : ФГБОУ ВО «КнАГУ», 2023. – С. 25–27.

87 Хрульков, В. Н. Особенности реализации нечетких регуляторов с применением универсальных платформ ПЛК / В. Н. Хрульков // Гагаринские чтения – 2020 : сборник тезисов докладов, Москва, 27 декабря 2019 года – 17 2020 года. – М. : Московский авиационный институт (национальный исследовательский университет), 2020. – С. 589.

88 Хрульков, В. Н. Применение аппарата нечеткой логики для управления гидравлическими системами при транспортировке сверхтяжелых карьерных самосвалов / В. Н. Хрульков, С. П. Черный // Производственные технологии будущего: от создания к внедрению : материалы IV Международной научно-практической конференции, 16 – 26 февраля 2021 года / редколлегия : С. И. Сухоруков (отв. ред.), А. С. Гудим, Н. Н. Любушкина. – Комсомольск-на-Амуре : ФГБОУ ВО «КнАГУ», 2021. – С. 72–75.

89 Хрульков, В. Н. Разработка модели системы управления подвеской многоопорной транспортной машины с применением теории мягких вычислений / В. Н. Хрульков // Известия Тульского государственного университета. Технические науки. – 2025. – № 2. С. 578–586

90 Хрульков, В. Н. Разработка пневматической экспериментальной установки для стабилизации грузов в пространстве с помощью нечеткого логического регулятора / В. Н. Хрульков, И. А. Иванова // Производственные технологии будущего: от создания к внедрению : материалы VI Международной научнопрактической конференции : в 2-х частях, Комсомольск-на-Амуре, 05 – 11 декабря 2022 года / редколлегия : С. И. Сухоруков (отв. ред.) [и др.]. Часть 1. – Комсомольск-на-Амуре : ФГБОУ ВО «КнАГУ», 2023. – С. 109–111.

91 Чебунин, А. Ф. Гидропривод транспортных и технологических машин : учебное пособие / А. Ф. Чебунин. – 2-е издание, исправленное. – Чита : ЗабГУ, 2012. – 135 с.

92 Черный, С. П. Особенности моделирования платформы при продольных и поперечных кренах / С. П. Черный, А. В. Охотников // Производственные технологии будущего: от создания к внедрению : материалы VI Международной научно-практической конференции молодых ученых : в 2-х частях, Комсомольск-на-Амуре, 05–11 декабря 2022 года / редколлегия : С. И. Сухоруков (отв. ред.) [и др.]. Часть 1. – Комсомольск-на-Амуре : ФГБОУ ВО «КнАГУ», 2023. – С. 128–130. – EDN QLYGZJ.

93 Шишмарев, В. Ю. Автоматика : учебник для среднего профессионального

образования / В. Ю. Шишмарев. – 2-е издание, исправленное и дополненное. – М. : Юрайт, 2022. – 280 с.

94 Штовба, С. Д. Введение в теорию нечетких множеств и нечеткую логику /С. Д. Штовба. – Винница : Континент-ПРИМ, 1997. – 246 с.

95 Эйкхофф, П. Основы идентификации систем управления / П. Эйкхофф. –
М.: Мир, 1975. – 683 с.

96 Abramov, S. Semi-Active Suspension System Simulation Using SIMULINK /
S. Abramov, S. Mannan, O. Durieux // International Journal of Engineering Systems
Modelling and Simulation. – 2009. – № 1. – Pp. 101–114.

97 Ang, K. H. PID control system analysis, design, and technology / K. H. Ang, G. Chong, Y. Li – Текст: непосредственный // IEEE Trans. on Control Systems Technology. – 2005. – Vol. 13. – № 4. – Pp. 559–576.

98 Bello, M. M. Active Vehicle Suspension Control using Full State-feedback Controller / M. M. Bello, A. A. Shafie, R. M. Khan // Advanced Materials Research. – 2015. – Vol. 1115. – Pp. 440–445.

99 Bello, M. M. Electro-hydraulic PID force control for nonlinear vehicle suspension system / M. M. Bello, A. A. Shafie, R. M. Khan // International Journal of Engineering Research & Technology. – 2015. – Vol. 4. – Is. 1. – Pp. 517–524.

100 Ву Хай Куан Моделирование и управление активной подвеской для модели <sup>1</sup>/<sub>4</sub> автомобиля / Ву Хай Куан, Нгуен Гуй Чыонг, Нгуен Чонг Дык // Конференция МАНФ-2020 «Наземные интеллектуальные транспортные средства и системы». – 2020. – С. 1–8.

101 Chandekar, B. Design of Electro-Hydraulic Active Suspension System for Four Wheel Vehicles / B. Chandekar, H. D. Lagdive // International Journal of Emerging Technology and Advanced Engineering. – 2014. – Vol.4. – Is. 4. – Pp. 885–889.

102 Dadios, E. Fuzzy Logic – Algorithms. Techniques and Implementations: монография / E. Dadios. – М. : ИнТех, 2012. – 294 с.

103 Fayyad, S. M. Constructing Control System for Active Suspension System /
S. M. Fayyad // Contemporary Engineering Sciences. – 2012. – Vol. 5. – No. 4. –
Pp. 189–200.

104 Hrovat, D. Survey of advanced suspension developments and related optimal control applications / D. Hrovat // Automatica. – 1997. – Vol. 33. – № 10. – Pp. 1781– 1817.

105 Hunt, K. J. Neural networks for control systems – a survey / K. J. Hunt, D. Sharbaro, R. Zbikowski, P. J. Gawthrop // Automatica. – Vol. 28. – № 6, 1992. – Pp. 1083–1112.

106 Kenan, Li Multibody dynamic simulation with Simscape: methods and examples / Li Kenan. – Polytechnic university of Turin, 2021. – 53 p.

107 Khrulkov, V. N Analysis of approaches to modelling the fuzzy control systems with extension of their functional capabilities / S. P. Cherny, V. I. Susdorf, A. V. Buzikaeva, V. N. Khrulkov // EAI Endorsed Transactions on Energy Web, 2021.

108 Khrulkov, V. N. High-speed energy-efficient power sources for electromechanical systems / S. A. Vasilchenko, S. P. Cherny, V. N. Khrulkov // Lecture Notes in Networks and Systems. – 2021. – Vol. 200. – P. 126-135. – DOI 10.1007/978-3-030-69421-0\_14. – EDN YVJQPI.

109 Khrulkov, V. N. Improving dynamic and energy characteristics of electromechanical systems with single-phase rectifiers / S. Vasilchenko, S. Cherny, V. Khrulkov // Proceedings - 2020 International Conference on Industrial Engineering, Applications and Manufacturing, ICIEAM 2020, Sochi, 18–22 мая 2020 года. – Sochi: Institute of Electrical and Electronics Engineers Inc., 2020. – P. 9111902. – DOI 10.1109/ICIEAM48468.2020.9111902. – EDN JJRMXQ.

110 Khrulkov, V. N. Intelligent Suspension Control System for Modular Trawl System / V. N. Khrulkov, S. P. Cherny // 2023 International Russian Automation Conference (RusAutoCon), 2023.

111 Khrulkov, V. Method of Implementing a Fuzzy Logic Controller by Hardware / V. Khrulkov, S. Vasilchenko, S. Cherny // 2020 International Multi-Conference on Industrial Engineering and Modern Technologies, FarEastCon 2020, Vladivostok, 06–09 октября 2020 года. – Vladivostok, 2020. – P. 9271539. – DOI 10.1109/FarEastCon50210.2020.9271539. – EDN RMKKKF.

112 Kosko, B. Fuzzy Systems as Universal Approximators / B. Kosko. - IEEE

Trans. Computers. – 1994. – Vol. 43. – № 11. – Pp.1329–1333.

113 Mamdani, E. H. Advances in the Linguistic Synthesis of Fuzzy Controller /
E. H. Mamdani // Int. J: Man-MachineStudies. – Vol. 8, 1976. – C. 669–678.

114 Mehta B. R. Industrial Process Automation Systems. 1st Edition. Design and Implementation / B. R. Mehta, Y. J. Reddy. – 2014. – 668 p.

115 Omar, M. A universal suspension test rig for electrohydraulic active and passive automotive suspension system / M. Omar, M. M. El-kassaby, W. Abdelghaffar // Alexandria Engineering Journal. – 2017. – Vol. 56. – Pp. 359–370.

116 Omar, M. Parametric numerical study of electrohydraulic active suspension performance against passive suspension / M. Omar, M. M. El-kassaby, W. Abdelghaffar // Alexandria Engineering Journal. – 2018. – Vol. 57. – Issue 4. – Pp. 3609–3614.

117 Senthil kumar, M. Analytical and experimental studies on active suspension system of light passenger vehicle to improve ride comfort / M. Senthil kumar, S. Vijayarangan // Mechanika.  $-2007. - Vol. 65. - N_{\odot} 3. - Pp. 34-41.$ 

118 Shows how to practically use Fuzzy Markup Language / Acampora, Giovanni; Loia, Vincenzo; Lee, Chang-Shing; Wang, Mei-Hui // On the Power of Fuzzy Markup Language. Studies in Fuzziness and Soft Computing. – 2013. – 296 p.

119 Stefano, D. On the Need of a Standard Language for Designing Fuzzy Systems / D. Stefano, N. Bruno // On the Power of Fuzzy Markup Language. Studies in Fuzziness and Soft Computing. -2013. -296 p.

120 Syropoulos, A. Theory of Fuzzy Computation: монография / A. Syropoulos. –Springer, 2014. – 170 p.

121 Venkateswarulu, E. The active suspension system with hydraulic actuator for half car model analysis and self-tuning with pid controllers / E. Venkateswarulu, N. Ramesh raju, G. Seshadri // International Journal of Research in Engineering and Technology. – 2014. – Vol. 3. – Pp. 415–421.

122 Wang, Jianjun Research and Improvement of the Hydraulic Suspension System for a Heavy Hydraulic Transport Vehicle / Jianjun Wang, Jingyi Zhao, Wenlei Li, Xing Jia and Peng Wei // Applied Sciences.  $-2020. - Vol. 10. - N_{\odot} 5220.$ 

123 Yin, C. Design and performance research of a hydro-pneumatic suspension

with variable damping and stiffness characteristics / C. Yin, X. Zhai, X. Sun, S. Wang, P. K. Wong // Journal of Mechanical Science and Technology. – 2022. – Vol. 36. – Pp. 4913–4923.

124 Zadeh, L. A. «Fuzzy algorithms» / L. A. Zadeh // Information and Control. – 1968. – № 12(2). – Pp. 94–102.

125 Zadeh, L. A. «Fuzzy sets» / L. A. Zadeh // Information and Control. – 1965. – № 8(3). – Pp. 338–353.

126 Zadeh, L. A. Fuzzy logic = computing with words. –IEEE. / L. A. Zadeh // Transaction on Fuzzy Systems. – Vol. 4. –  $N_{2}$  2. –1996. – Pp. 103–111.

127 Zadeh, L. A. The role of fuzzy logic in the management of uncertainty in expert systems. Fuzzy Sets a. Systems / L. A. Zadeh. – Perg. Press : Automation. – Vol.  $11. - N_{2} 3$ , 1983. – Pp. 199–227.

128 Ziegler, J. G. Optimum settings for automatic controllers / J. G. Ziegler, N. B. Nichols // Trans. ASME. – 1942. – Pp. 759–768.

129 Zimmerman, H. J. Fuzzy set theory and its application / H. J. Zimmerman. – London : Kluwer Academic Publishers, 1994.

## Приложение А

## (обязательное)

#### Акты внедрения

ФГБОУ ВО «Комсомольский-на-Амуре государственный университет»



АКТ о внедрении в учебный процесс

Комиссия в составе председателя - декана факультета энергетики и управления, доцента Гудима А.С., и членов комиссии: профессора кафедры ЭПАПУ, профессора Соловьева В.А., доцента кафедры ЭПАПУ, доцента Стельмащука С.В., доцента кафедры ЭПАПУ, доцента Сухорукова С.И. составили настоящий акт в том, что с 2021 года в Комсомольском-на-Амуре государственном университете внедрены в учебный процесс кафедры «Электропривод и автоматизация промышленных установок» для направлений подготовки 15.03.06 «Мехатроника и робототехника», профиль «Робототехнические комплексы и системы»; 27.03.04 «Управление в технических системах», профиль «Автоматизация и управление технологическими процессами»; 27.04.04 «Управление в технических системах», профиль «Управление и информатика в технических системах» результаты диссертационной работы Хрулькова В.Н.

Вид внедренных результатов:

1. Алгоритмы нечеткого многокаскадного управления для гидравлических систем;

 Модель многокаскадного нечеткого логического регулятора для мехатронных модулей синтезирующего сигнал управления вложенными каскадами на основе операторов произведения и суммы.

Результаты используются в курсах «Интеллектуальные технологии в управлении техническими системами», «Искусственный интеллект в задачах управления», «Технологии роботизированного производства», а также при курсовом проектировании и при выполнении ВКР.

Председатель комиссии: Члены комиссии:

/А.С. Гудим/ el /В.А. Соловьев / +С.В. Стельмащук/ /С.И. Сухоруков / /Е.Е. Поздеева/

Начальник УМУ

#### АКТ

Об использовании научных результатов диссертационной работы Хрулькова Владимира Николаевича на тему «Разработка и исследование интеллектуальной системы управления комбинированным траловым механизмом для перемещения высокотоннажных крупногабаритных грузов»

Результаты диссертационной работы Хрулькова Владимира Николаевича на тему «Разработка и исследование интеллектуальной системы управления комбинированным траловым механизмом для перемещения высокотоннажных крупногабаритных грузов», виде в алгоритмов управления, использовались на Комсомольской дистанции электроснабжения Дальневосточной дирекции по энергообеспечению ОАО «РЖД» при перемещении крупногабаритных высокотоннажных объектов, таких как мостовые конструкции, так же рельсо-шпальные решетки а железнодорожного пути при загрузке путеукладочных машин с применением транспортировочного модуля. Использование теоретических и практических данных, полученных в ходе диссертационного исследования, для элементов и алгоритмов интеллектуальных систем управления позволило адаптировать методы принятия решений по оптимизации перемещения таких крупногабаритных высокотоннажных объектов в ходе реконструкции объектов ОАО «РЖД».

Заместитель начальника Комсомольской дистанции электроснабжения ОАО «РЖД» (должность)

Барышев Сергей Валерьевич (ФИО) (подпись)

## Приложение Б

## (обязательное)

## Свидетельства регистраций программ для ЭВМ







СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2023617229

## «ПРОГРАММА РЕАЛИЗАЦИИ НЕЧЕТКОГО РЕГУЛЯТОРА ДЛЯ СИСТЕМЫ УПРАВЛЕНИЯ МАЛОМОЩНЫМ ЭЛЕКТРОПРИВОДОМ»

Правообладатель: Федеральное государственное бюджетное образовательное учреждение высшего образования «Комсомольский-на-Амуре государственный университет» (ФГБОУ ВО «КнАГУ») (RU)

Авторы: Бузикаева Алина Валерьевна (RU), Черный Сергей Петрович (RU), Хрульков Владимир Николаевич (RU)

**密密路路路路路路路路路路路路路路路路路路路路路路**路路路路



密

密

路

密

斑

斑

密

斑

斑

肉

路路

密

密

密

路路路

密

密

密

密

路

密

路路

密

密

密

密

路路路路

密

密

密

密

路

路

段

敬敬

密

密

磁磁磁磁磁

Заявка № 2023616075

Дата поступления **28 марта 2023 г.** Дата государственной регистрации в Реестре программ для ЭВМ *06 апреля 2023 г.* 

> Руководитель Федеральной службы по интеллектуальной собственности

Ю.С. Зубов

路路路路路路



## Приложение В

## (обязательное)

## Код программы многокаскадного нечеткого логического регулятора,

## формирующий уставки внутренних модулей за счет суммирования значений

#### сигналов управления

#include "Code MNLR v3.h" #include "rtwtypes.h" #include "Code MNLR v3 private.h" #include <math.h> #include <string.h> #include <stddef.h> /\* Block signals (default storage) \*/ B\_Code\_MNLR\_v3\_T Code\_MNLR\_v3\_B; /\* Continuous states \*/ X\_Code\_MNLR\_v3\_T Code\_MNLR\_v3\_X; /\* Block states (default storage) \*/ DW\_Code\_MNLR\_v3\_T Code\_MNLR\_v3\_DW; /\* Mass Matrices \*/ MassMatrix\_Code\_MNLR\_v3\_T Code\_MNLR\_v3\_MassMatrix; /\* Real-time model \*/ static RT\_MODEL\_Code\_MNLR\_v3\_T Code\_MNLR\_v3\_M\_; RT MODEL Code MNLR v3 T \*const Code MNLR v3 M = &Code MNLR v3 M ; /\* Forward declaration for local functions \*/ static real\_T Code\_MNLR\_v3\_trimf(real\_T x, const real\_T params[3]); /\* \* Time delay interpolation routine \* \* The linear interpolation is performed using the formula: \* (t2 - tMinusDelay) (tMinusDelay - t1) \* u(t) = ----- \* u1 + ----- \* u2\* (t2 - t1) (t2 - t1)\*/ real T rt TDelayInterpolate( /\* tMinusDelay = currentSimTime - delay \*/ real\_T tMinusDelay, real\_T tStart, real\_T \*uBuf, int\_T bufSz, int T \*lastIdx, int T oldestIdx, int\_T newIdx, real T initOutput, boolean\_T discrete, boolean\_T minorStepAndTAtLastMajorOutput) { int\_T i; real\_T yout, t1, t2, u1, u2; real  $T^*$  tBuf = uBuf + bufSz; /\* \* If there is only one data point in the buffer, this data point must be \* the t=0 and tMinusDelay > t0, it ask for something unknown. The best \* guess if initial output as well

```
if ((\text{newIdx} == 0) \&\& (\text{oldestIdx} == 0) \&\& (\text{tMinusDelay} > \text{tStart}))
 return initOutput;
/*
* If tMinusDelay is less than zero, should output initial value
*/
if (tMinusDelay <= tStart)
 return initOutput;
/* For fixed buffer extrapolation:
* if tMinusDelay is small than the time at oldestIdx, if discrete, output
* tailptr value, else use tailptr and tailptr+1 value to extrapolate
* It is also for fixed buffer. Note: The same condition can happen for transport delay block where
* use tStart and and t[tail] other than using t[tail] and t[tail+1].
* See below
*/
if ((tMinusDelay <= tBuf[oldestIdx])) {
 if (discrete) {
  return(uBuf[oldestIdx]);
 } else {
  int T tempIdx = oldestIdx + 1;
  if (oldestIdx == bufSz-1)
   tempIdx = 0;
  t1=tBuf[oldestIdx];
  t2 = tBuf[tempIdx];
  u1= uBuf[oldestIdx];
  u2= uBuf[tempIdx];
  if (t^2 == t^1) {
   if (tMinusDelay \geq t2) {
     yout = u2;
    } else {
     yout = u1;
    ł
  } else {
   real_T f1 = (t2-tMinusDelay) / (t2-t1);
   real_T f2 = 1.0 - f1;
   /*
    * Use Lagrange's interpolation formula. Exact outputs at t1, t2.
   yout = f1*u1 + f2*u2;
  }
  return yout;
 }
}
/*
* When block does not have direct feedthrough, we use the table of
* values to extrapolate off the end of the table for delays that are less
* than 0 (less then step size). This is not completely accurate. The
* chain of events is as follows for a given time t. Major output - look
* in table. Update - add entry to table. Now, if we call the output at
* time t again, there is a new entry in the table. For very small delays,
* this means that we will have a different answer from the previous call
* to the output fcn at the same time t. The following code prevents this
* from happening.
*/
if (minorStepAndTAtLastMajorOutput) {
 /* pretend that the new entry has not been added to table */
 if (newIdx != 0) {
  if (*lastIdx == newIdx) {
```

\*/

```
(*lastIdx)--;
  }
  newIdx--;
 } else {
  if (*lastIdx == newIdx) {
    *lastIdx = bufSz-1;
  }
  newIdx = bufSz - 1;
 }
}
i = *lastIdx;
if (tBuf[i] < tMinusDelay) {
 /* Look forward starting at last index */
 while (tBuf[i] < tMinusDelay) {
  /* May occur if the delay is less than step-size - extrapolate */
  if (i == newIdx)
   break;
  i = (i < (bufSz-1))?(i+1): 0;/* move through buffer */
 }
} else {
 /*
  * Look backwards starting at last index which can happen when the
  * delay time increases.
  */
 while (tBuf[i] >= tMinusDelay) {
  /*
   * Due to the entry condition at top of function, we
   * should never hit the end.
   */
  i = (i > 0)? i-1 : (bufSz-1); /* move through buffer */
 }
 i = (i < (bufSz-1))?(i+1):0;
}
*lastIdx = i;
if (discrete) {
 /*
  * tempEps = 128 * eps;
  * localEps = max(tempEps, tempEps*fabs(tBuf[i]))/2;
  */
 double tempEps = (DBL_EPSILON) * 128.0;
 double localEps = tempEps * fabs(tBuf[i]);
 if (tempEps > localEps) {
  localEps = tempEps;
 }
 localEps = localEps / 2.0;
 if (tMinusDelay >= (tBuf[i] - localEps)) {
  yout = uBuf[i];
 } else {
  if (i == 0) {
   yout = uBuf[bufSz-1];
  } else {
   yout = uBuf[i-1];
  }
 }
} else {
 if (i == 0) {
  t1 = tBuf[bufSz-1];
  u1 = uBuf[bufSz-1];
```

```
} else {
   t1 = tBuf[i-1];
   u1 = uBuf[i-1];
  t2 = tBuf[i];
  u^2 = uBuf[i];
  if (t^2 == t^1) {
   if (tMinusDelay \geq t2) {
    yout = u2;
   } else {
    yout = u1;
   }
  } else {
   real_T f1 = (t2-tMinusDelay) / (t2-t1);
   real T f_2 = 1.0 - f_1;
   /*
    * Use Lagrange's interpolation formula. Exact outputs at t1, t2.
   yout = f1*u1 + f2*u2;
  }
 }
 return(yout);
/* ForcingFunction for root system: '<Root>' */
void Code_MNLR_v3_forcingfunction(void)
{
 NeslSimulationData *simulationData;
 NeuDiagnosticManager *diagnosticManager;
 NeuDiagnosticTree *diagnosticTree:
 XDot_Code_MNLR_v3_T *_rtXdot;
 char *msg;
 real_T tmp_0[132];
 real_T time;
 int32 T tmp 2;
 int_T tmp_1[34];
 boolean T tmp;
 _rtXdot = ((XDot_Code_MNLR_v3_T *) Code_MNLR_v3_M->derivs);
 /* ForcingFunction for Integrator: '<S214>/Integrator' */
 _rtXdot->Integrator_CSTATE = Code_MNLR_v3_B.IntegralGain_en;
 /* ForcingFunction for Integrator: '<S209>/Filter' */
 _rtXdot->Filter_CSTATE = Code_MNLR_v3_B.FilterCoefficient;
 /* ForcingFunction for Integrator: '<S886>/Integrator' */
 _rtXdot->Integrator_CSTATE_p = Code_MNLR_v3_B.IntegralGain_k;
 /* ForcingFunction for Integrator: '<S881>/Filter' */
 _rtXdot->Filter_CSTATE_p = Code_MNLR_v3_B.FilterCoefficient_n;
 /* ForcingFunction for Integrator: '<S262>/Integrator' */
 _rtXdot->Integrator_CSTATE_n = Code_MNLR_v3_B.IntegralGain_b;
 /* ForcingFunction for Integrator: '<S257>/Filter' */
 _rtXdot->Filter_CSTATE_i = Code_MNLR_v3_B.FilterCoefficient_l;
 /* ForcingFunction for Integrator: '<S310>/Integrator' */
 _rtXdot->Integrator_CSTATE_j = Code_MNLR_v3_B.IntegralGain_c;
 /* ForcingFunction for Integrator: '<S305>/Filter' */
 _rtXdot->Filter_CSTATE_k = Code_MNLR_v3_B.FilterCoefficient m;
 /* ForcingFunction for Integrator: '<S358>/Integrator' */
 rtXdot->Integrator CSTATE m = Code MNLR v3 B.IntegralGain o3;
 /* ForcingFunction for Integrator: '<S353>/Filter' */
 _rtXdot->Filter_CSTATE_h = Code_MNLR_v3_B.FilterCoefficient_a;
 /* ForcingFunction for Integrator: '<S502>/Integrator' */
```

\_rtXdot->Integrator\_CSTATE\_1 = Code\_MNLR\_v3\_B.IntegralGain\_f; /\* ForcingFunction for Integrator: '<S497>/Filter' \*/ \_rtXdot->Filter\_CSTATE\_a = Code\_MNLR\_v3\_B.FilterCoefficient p; /\* ForcingFunction for Integrator: '<S406>/Integrator' \*/ \_rtXdot->Integrator\_CSTATE\_ma = Code\_MNLR\_v3\_B.IntegralGain\_em; /\* ForcingFunction for Integrator: '<S401>/Filter' \*/ rtXdot->Filter CSTATE g = Code MNLR v3 B.FilterCoefficient i; /\* ForcingFunction for Integrator: '<S550>/Integrator' \*/ \_rtXdot->Integrator\_CSTATE\_is = Code\_MNLR\_v3\_B.IntegralGain\_a; /\* ForcingFunction for Integrator: '<S545>/Filter' \*/ \_rtXdot->Filter\_CSTATE\_gy = Code\_MNLR\_v3\_B.FilterCoefficient\_ab; /\* ForcingFunction for Integrator: '<S454>/Integrator' \*/ rtXdot->Integrator CSTATE a = Code MNLR v3 B.IntegralGain g; /\* ForcingFunction for Integrator: '<S449>/Filter' \*/ \_rtXdot->Filter\_CSTATE\_ay = Code\_MNLR\_v3\_B.FilterCoefficient\_e; /\* ForcingFunction for Integrator: '<S166>/Integrator' \*/ \_rtXdot->Integrator\_CSTATE\_f = Code\_MNLR\_v3\_B.IntegralGain\_i; /\* ForcingFunction for Integrator: '<S161>/Filter' \*/ \_rtXdot->Filter\_CSTATE\_n = Code\_MNLR\_v3\_B.FilterCoefficient\_ni; /\* ForcingFunction for Integrator: '<S598>/Integrator' \*/ \_rtXdot->Integrator\_CSTATE\_i = Code\_MNLR\_v3\_B.IntegralGain\_m; /\* ForcingFunction for Integrator: '<S593>/Filter' \*/ rtXdot->Filter CSTATE o = Code MNLR v3 B.FilterCoefficient k; /\* ForcingFunction for Integrator: '<S646>/Integrator' \*/ \_rtXdot->Integrator\_CSTATE\_k = Code\_MNLR\_v3\_B.IntegralGain e; /\* ForcingFunction for Integrator: '<S641>/Filter' \*/ \_rtXdot->Filter\_CSTATE\_pb = Code\_MNLR\_v3\_B.FilterCoefficient\_me; /\* ForcingFunction for TransferFcn: '<S103>/Transfer Fcn7' \*/ \_rtXdot->TransferFcn7\_CSTATE = 0.0; \_rtXdot->TransferFcn7\_CSTATE += -0.0 \* Code\_MNLR\_v3\_X.TransferFcn7\_CSTATE; \_rtXdot->TransferFcn7\_CSTATE += Code\_MNLR\_v3\_B.Gain6; /\* ForcingFunction for Integrator: '<S742>/Integrator' \*/ \_rtXdot->Integrator\_CSTATE\_jt = Code\_MNLR\_v3\_B.IntegralGain\_o; /\* ForcingFunction for Integrator: '<S737>/Filter' \*/ \_rtXdot->Filter\_CSTATE\_d = Code\_MNLR\_v3\_B.FilterCoefficient i2; /\* ForcingFunction for Integrator: '<S694>/Integrator' \*/ \_rtXdot->Integrator\_CSTATE\_kg = Code\_MNLR\_v3\_B.IntegralGain\_d; /\* ForcingFunction for Integrator: '<S689>/Filter' \*/ rtXdot->Filter CSTATE i4 = Code MNLR v3 B.FilterCoefficient c; /\* ForcingFunction for Integrator: '<S790>/Integrator' \*/ \_rtXdot->Integrator\_CSTATE\_j2 = Code\_MNLR\_v3\_B.IntegralGain\_l; /\* ForcingFunction for Integrator: '<S785>/Filter' \*/ \_rtXdot->Filter\_CSTATE\_j = Code\_MNLR\_v3\_B.FilterCoefficient\_c0; /\* ForcingFunction for Integrator: '<S838>/Integrator' \*/ rtXdot->Integrator CSTATE ft = Code MNLR v3 B.IntegralGain; /\* ForcingFunction for Integrator: '<S833>/Filter' \*/ \_rtXdot->Filter\_CSTATE\_ku = Code\_MNLR\_v3\_B.FilterCoefficient\_al; /\* ForcingFunction for SimscapeExecutionBlock: '<S954>/STATE\_1' \*/ simulationData = (NeslSimulationData \*)Code\_MNLR\_v3\_DW.STATE\_1\_SimData; time = Code MNLR v3 M->Timing.t[0]; simulationData->mData->mTime.mN = 1; simulationData->mData->mTime.mX = &time; simulationData->mData->mContStates.mN = 245; simulationData->mData->mContStates.mX = &Code MNLR v3 X.Code MNLR v3Double Acting Hydra[0]; simulationData->mData->mDiscStates.mN = 0; simulationData->mData->mDiscStates.mX = &Code\_MNLR\_v3\_DW.STATE\_1\_Discrete; simulationData->mData->mModeVector.mN = 194;

simulationData->mData->mModeVector.mX = &Code\_MNLR\_v3\_DW.STATE\_1\_Modes[0]; tmp = false;simulationData->mData->mFoundZcEvents = tmp; simulationData->mIsMajorTimeStep = rtmIsMajorTimeStep(Code\_MNLR v3 M); tmp = false:simulationData->mIsSolverAssertCheck = tmp; simulationData->mData->mIsSolverCheckingCIC = false; tmp = rtsiIsSolverComputingJacobian(&Code\_MNLR\_v3\_M->solverInfo); simulationData->mData->mIsComputingJacobian = tmp; simulationData->mData->mIsEvaluatingF0 = false; simulationData->mData->mIsSolverRequestingReset = false; simulationData->mData->mIsModeUpdateTimeStep = rtsiIsModeUpdateTimeStep (&Code MNLR v3 M->solverInfo);  $tmp_1[0] = 0;$ tmp 0[0] = Code MNLR v3 B.INPUT 12 1 1[0]; $tmp_0[1] = Code_MNLR_v3_B.INPUT_12_1_1[1];$  $tmp_0[2] = Code_MNLR_v3_B.INPUT_12_1_1[2];$  $tmp_0[3] = Code_MNLR_v3_B.INPUT_12_1_1[3];$  $tmp_1[1] = 4;$ tmp 0[4] = Code MNLR v3 B.INPUT 1 1 1[0];tmp\_0[5] = Code\_MNLR\_v3\_B.INPUT\_1\_1\_1[1];  $tmp_0[6] = Code_MNLR_v3_B.INPUT_1_1_1[2];$ tmp 0[7] = Code MNLR v3 B.INPUT 1 1 1[3]; $tmp_1[2] = 8;$  $tmp_0[8] = Code_MNLR_v3_B.INPUT_13_1_1[0];$ tmp\_0[9] = Code\_MNLR\_v3\_B.INPUT\_13\_1\_1[1];  $tmp_0[10] = Code_MNLR_v3_B.INPUT_13_1_1[2];$ tmp 0[11] = Code MNLR v3 B.INPUT 13 1 1[3]; $tmp_1[3] = 12;$  $tmp_0[12] = Code_MNLR_v3_B.INPUT_15_1_1[0];$  $tmp_0[13] = Code_MNLR_v3_B.INPUT_15_1_1[1];$  $tmp_0[14] = Code_MNLR_v3_B.INPUT_15_1_1[2];$  $tmp_0[15] = Code_MNLR_v3_B.INPUT_15_1_1[3];$  $tmp_1[4] = 16;$ tmp\_0[16] = Code\_MNLR\_v3\_B.INPUT\_21\_1\_1[0]; tmp 0[17] = Code MNLR v3 B.INPUT 21 1 1[1]; $tmp_0[18] = Code_MNLR_v3_B.INPUT_21_1_1[2];$  $tmp_0[19] = Code_MNLR_v3_B.INPUT_21_1_1[3];$ tmp 1[5] = 20; $tmp_0[20] = Code_MNLR_v3_B.INPUT_24_1_1[0];$ tmp\_0[21] = Code\_MNLR\_v3\_B.INPUT\_24\_1\_1[1]; tmp\_0[22] = Code\_MNLR\_v3\_B.INPUT\_24\_1\_1[2]; tmp\_0[23] = Code\_MNLR\_v3\_B.INPUT\_24\_1\_1[3]; tmp 1[6] = 24;tmp 0[24] = Code MNLR v3 B.INPUT 18 1 1[0]; $tmp_0[25] = Code_MNLR_v3_B.INPUT_18_1_1[1];$  $tmp_0[26] = Code_MNLR_v3_B.INPUT_18_1_1[2];$ tmp\_0[27] = Code\_MNLR\_v3\_B.INPUT\_18\_1\_1[3];  $tmp_1[7] = 28;$  $tmp_0[28] = Code_MNLR_v3_B.INPUT_26_1_1[0];$ tmp\_0[29] = Code\_MNLR\_v3\_B.INPUT\_26\_1\_1[1]; tmp 0[30] = Code MNLR v3 B.INPUT 26 1 1[2]; $tmp_0[31] = Code_MNLR_v3_B.INPUT_26_1_1[3];$ tmp 1[8] = 32;tmp 0[32] = Code MNLR v3 B.INPUT 29 1 1[0]; $tmp_0[33] = Code_MNLR_v3_B.INPUT_29_1_1[1];$ tmp\_0[34] = Code\_MNLR\_v3\_B.INPUT\_29\_1\_1[2]; tmp\_0[35] = Code\_MNLR\_v3\_B.INPUT\_29\_1\_1[3];

 $tmp_1[9] = 36;$ tmp 0[36] = Code MNLR v3 B.INPUT 31 1 1[0]; $tmp_0[37] = Code_MNLR_v3_B.INPUT_31_1_1[1];$ tmp\_0[38] = Code\_MNLR\_v3\_B.INPUT\_31\_1\_1[2];  $tmp_0[39] = Code_MNLR_v3_B.INPUT_31_1_1[3];$  $tmp_1[10] = 40;$ tmp 0[40] = Code MNLR v3 B.INPUT 2 1 1[0]; $tmp_0[41] = Code_MNLR_v3_B.INPUT_2_1_1[1];$ tmp\_0[42] = Code\_MNLR\_v3\_B.INPUT\_2\_1\_1[2];  $tmp_0[43] = Code_MNLR_v3_B.INPUT_2_1_1[3];$  $tmp_1[11] = 44;$ tmp 0[44] = Code MNLR v3 B.INPUT 28 1 1[0];tmp 0[45] = Code MNLR v3 B.INPUT 28 1 1[1]; $tmp_0[46] = Code_MNLR_v3_B.INPUT_28_1_1[2];$  $tmp_0[47] = Code_MNLR_v3_B.INPUT_28_1_1[3];$  $tmp_1[12] = 48;$  $tmp_0[48] = Code_MNLR_v3_B.INPUT_5_1_1[0];$  $tmp_0[49] = Code_MNLR_v3_B.INPUT_5_1_1[1];$  $tmp_0[50] = Code_MNLR_v3_B.INPUT_5_1_1[2];$ tmp 0[51] = Code MNLR v3 B.INPUT 5 1 1[3];tmp 1[13] = 52; $tmp_0[52] = Code_MNLR_v3_B.INPUT_7_1_1[0];$ tmp 0[53] = Code MNLR v3 B.INPUT 7 1 1[1]; $tmp_0[54] = Code_MNLR_v3_B.INPUT_7_1_1[2];$  $tmp_0[55] = Code_MNLR_v3_B.INPUT_7_1_1[3];$ tmp 1[14] = 56;tmp\_0[56] = Code\_MNLR\_v3\_B.INPUT\_9\_1\_1[0]; tmp 0[57] = Code MNLR v3 B.INPUT 9 1 1[1];tmp  $0[58] = Code_MNLR_v3_B.INPUT_9_1_1[2];$ tmp\_0[59] = Code\_MNLR\_v3\_B.INPUT\_9\_1\_1[3]; tmp 1[15] = 60; $tmp_0[60] = Code_MNLR_v3_B.INPUT_17_1_1[0];$  $tmp_0[61] = Code_MNLR_v3_B.INPUT_17_1_1[1];$ tmp 0[62] = Code MNLR v3 B.INPUT 17 1 1[2];tmp\_0[63] = Code\_MNLR\_v3\_B.INPUT\_17\_1\_1[3]; tmp 1[16] = 64; $tmp_0[64] = Code_MNLR_v3_B.INPUT_3_1_1[0];$ tmp\_0[65] = Code\_MNLR\_v3\_B.INPUT\_3\_1\_1[1]; tmp 0[66] = Code MNLR v3 B.INPUT 3 1 1[2]; $tmp_0[67] = Code_MNLR_v3_B.INPUT_3_1_1[3];$  $tmp_1[17] = 68;$  $tmp_0[68] = Code_MNLR_v3_B.INPUT_4_1_1[0];$ tmp\_0[69] = Code\_MNLR\_v3\_B.INPUT\_4\_1\_1[1]; tmp 0[70] = Code MNLR v3 B.INPUT 4 1 1[2]; $tmp_0[71] = Code_MNLR_v3_B.INPUT_4_1_1[3];$  $tmp_1[18] = 72;$ tmp 0[72] = Code MNLR v3 B.INPUT 6 1 1[0]; $tmp_0[73] = Code_MNLR_v3_B.INPUT_6_1_1[1];$  $tmp_0[74] = Code_MNLR_v3_B.INPUT_6_1_1[2];$  $tmp_0[75] = Code_MNLR_v3_B.INPUT_6_1_1[3];$ tmp\_1[19] = 76; tmp 0[76] = Code MNLR v3 B.INPUT 8 1 1[0]; $tmp_0[77] = Code_MNLR_v3_B.INPUT_8_1_1[1];$ tmp\_0[78] = Code\_MNLR\_v3\_B.INPUT\_8\_1\_1[2]; tmp 0[79] = Code MNLR v3 B.INPUT 8 1 1[3]; $tmp_1[20] = 80;$  $tmp_0[80] = Code_MNLR_v3_B.INPUT_10_1_1[0];$  $tmp_0[81] = Code_MNLR_v3_B.INPUT_10_1_1[1];$
tmp\_0[82] = Code\_MNLR\_v3\_B.INPUT\_10\_1\_1[2];  $tmp_0[83] = Code_MNLR_v3_B.INPUT_10_1_1[3];$  $tmp_1[21] = 84;$  $tmp_0[84] = Code_MNLR_v3_B.INPUT_11_1_1[0];$  $tmp_0[85] = Code_MNLR_v3_B.INPUT_11_1[1];$  $tmp_0[86] = Code_MNLR_v3_B.INPUT_11_1_1[2];$  $tmp_0[87] = Code_MNLR_v3_B.INPUT_11_1[3];$ tmp\_1[22] = 88;  $tmp_0[88] = Code_MNLR_v3_B.INPUT_14_1_1[0];$ tmp\_0[89] = Code\_MNLR\_v3\_B.INPUT\_14\_1\_1[1];  $tmp_0[90] = Code_MNLR_v3_B.INPUT_14_1_1[2];$ tmp 0[91] = Code MNLR v3 B.INPUT 14 1 1[3];tmp 1[23] = 92; $tmp_0[92] = Code_MNLR_v3_B.INPUT_16_1_1[0];$  $tmp_0[93] = Code_MNLR_v3_B.INPUT_16_1_1[1];$  $tmp_0[94] = Code_MNLR_v3_B.INPUT_16_1_1[2];$  $tmp_0[95] = Code_MNLR_v3_B.INPUT_16_1_1[3];$  $tmp_1[24] = 96;$  $tmp_0[96] = Code_MNLR_v3_B.INPUT_19_1_1[0];$ tmp 0[97] = Code MNLR v3 B.INPUT 19 1 1[1];tmp\_0[98] = Code\_MNLR\_v3\_B.INPUT\_19\_1\_1[2];  $tmp_0[99] = Code_MNLR_v3_B.INPUT_19_1_1[3];$ tmp 1[25] = 100; $tmp_0[100] = Code_MNLR_v3_B.INPUT_20_1_1[0];$ tmp\_0[101] = Code\_MNLR\_v3\_B.INPUT\_20\_1\_1[1]; tmp\_0[102] = Code\_MNLR\_v3\_B.INPUT\_20\_1\_1[2]; tmp\_0[103] = Code\_MNLR\_v3\_B.INPUT\_20\_1\_1[3]; tmp 1[26] = 104;tmp 0[104] = Code MNLR v3 B.INPUT 22 1 1[0];tmp\_0[105] = Code\_MNLR\_v3\_B.INPUT\_22\_1\_1[1]; tmp\_0[106] = Code\_MNLR\_v3\_B.INPUT\_22\_1\_1[2]; tmp\_0[107] = Code\_MNLR\_v3\_B.INPUT\_22\_1\_1[3]; tmp 1[27] = 108;tmp 0[108] = Code MNLR v3 B.INPUT 25 1 1[0];tmp\_0[109] = Code\_MNLR\_v3\_B.INPUT\_25\_1\_1[1]; tmp 0[110] = Code MNLR v3 B.INPUT 25 1 1[2]; $tmp_0[111] = Code_MNLR_v3_B.INPUT_25_1_1[3];$ tmp 1[28] = 112;tmp 0[112] = Code MNLR v3 B.INPUT 23 1 1[0]; $tmp_0[113] = Code_MNLR_v3_B.INPUT_23_1_1[1];$ tmp\_0[114] = Code\_MNLR\_v3\_B.INPUT\_23\_1\_1[2]; tmp\_0[115] = Code\_MNLR\_v3\_B.INPUT\_23\_1\_1[3]; tmp\_1[29] = 116; tmp 0[116] = Code MNLR v3 B.INPUT 27 1 1[0];tmp 0[117] = Code MNLR v3 B.INPUT 27 1 1[1];tmp\_0[118] = Code\_MNLR\_v3\_B.INPUT\_27\_1\_1[2];  $tmp_0[119] = Code_MNLR_v3_B.INPUT_27_1_1[3];$  $tmp_1[30] = 120;$  $tmp_0[120] = Code_MNLR_v3_B.INPUT_30_1_1[0];$  $tmp_0[121] = Code_MNLR_v3_B.INPUT_30_1_1[1];$ tmp\_0[122] = Code\_MNLR\_v3\_B.INPUT\_30\_1\_1[2]; tmp 0[123] = Code MNLR v3 B.INPUT 30 1 1[3];tmp 1[31] = 124;tmp\_0[124] = Code\_MNLR\_v3\_B.INPUT\_32\_1\_1[0]; tmp 0[125] = Code MNLR v3 B.INPUT 32 1 1[1];tmp\_0[126] = Code\_MNLR\_v3\_B.INPUT\_32\_1\_1[2]; tmp\_0[127] = Code\_MNLR\_v3\_B.INPUT\_32\_1\_1[3];  $tmp_1[32] = 128;$ 

tmp\_0[128] = Code\_MNLR\_v3\_B.INPUT\_33\_1\_1[0]; tmp 0[129] = Code MNLR v3 B.INPUT 33 1 1[1];tmp\_0[130] = Code\_MNLR\_v3\_B.INPUT\_33\_1\_1[2];  $tmp_0[131] = Code_MNLR_v3_B.INPUT_33_1_1[3];$ tmp 1[33] = 132;simulationData->mData->mInputValues.mN = 132; simulationData->mData->mInputValues.mX = &tmp 0[0]; simulationData->mData->mInputOffsets.mN = 34; simulationData->mData->mInputOffsets.mX = &tmp\_1[0]; simulationData->mData->mDx.mN = 245; simulationData->mData->mDx.mX = &\_rtXdot->Code\_MNLR\_v3Double\_Acting\_Hydra[0]; diagnosticManager = (NeuDiagnosticManager \*)Code MNLR v3 DW.STATE 1 DiagMgr; diagnosticTree = neu diagnostic manager get initial tree(diagnosticManager); tmp\_2 = ne\_simulator\_method((NeslSimulator \*)Code\_MNLR\_v3\_DW.STATE\_1\_Simulator, NESL SIM FORCINGFUNCTION, simulationData, diagnosticManager); if (tmp\_2 != 0) { tmp = error\_buffer\_is\_empty(rtmGetErrorStatus(Code\_MNLR\_v3\_M)); if (tmp) { msg = rtw\_diagnostics\_msg(diagnosticTree); rtmSetErrorStatus(Code MNLR v3 M, msg); } } /\* End of ForcingFunction for SimscapeExecutionBlock: '<S954>/STATE\_1' \*/ /\* ForcingFunction for TransferFcn: '<S103>/Transfer Fcn6' \*/ \_rtXdot->TransferFcn6\_CSTATE = 0.0; \_rtXdot->TransferFcn6\_CSTATE += -30.120474261342469 \* Code MNLR v3 X.TransferFcn6 CSTATE; \_rtXdot->TransferFcn6\_CSTATE += Code\_MNLR\_v3\_B.Sum6\_j; /\* ForcingFunction for TransferFcn: '<S103>/Transfer Fcn4' \*/ \_rtXdot->TransferFcn4\_CSTATE = 0.0; \_rtXdot->TransferFcn4\_CSTATE += -0.0 \* Code\_MNLR\_v3\_X.TransferFcn4\_CSTATE; rtXdot->TransferFcn4 CSTATE += Code MNLR v3 B.Sum5 e; /\* ForcingFunction for TransferFcn: '<S103>/Transfer Fcn5' \*/ \_rtXdot->TransferFcn5\_CSTATE = 0.0; \_rtXdot->TransferFcn5\_CSTATE += -142.85714285714286 \* Code MNLR v3 X.TransferFcn5 CSTATE; \_rtXdot->TransferFcn5\_CSTATE += Code\_MNLR\_v3\_B.TransferFcn4; /\* MassMatrix for root system: '<Root>' \*/ void Code\_MNLR\_v3\_massmatrix(void) NeslSimulationData \*simulationData; NeuDiagnosticManager \*diagnosticManager; NeuDiagnosticTree \*diagnosticTree; char \*msg; real\_T tmp\_0[132]; real T time; real\_T \*tmp\_2; real T \*tmp 3; int32\_T tmp\_4; int\_T tmp\_1[34]; boolean T tmp; /\* MassMatrix for SimscapeExecutionBlock: '<S954>/STATE\_1' \*/ simulationData = (NeslSimulationData \*)Code\_MNLR\_v3\_DW.STATE\_1\_SimData; time = Code\_MNLR\_v3\_M->Timing.t[0];

```
simulationData->mData->mTime.mN = 1;
simulationData->mData->mTime.mX = &time;
simulationData->mData->mContStates.mN = 245;
simulationData->mData->mContStates.mX =
 &Code MNLR v3 X.Code MNLR v3Double Acting Hydra[0];
simulationData->mDiscStates.mN = 0;
simulationData->mData->mDiscStates.mX = &Code MNLR v3 DW.STATE 1 Discrete;
simulationData->mData->mModeVector.mN = 194;
simulationData->mData->mModeVector.mX = &Code_MNLR_v3_DW.STATE_1_Modes[0];
tmp = false;
simulationData->mData->mFoundZcEvents = tmp;
simulationData->mData->mIsMajorTimeStep = rtmIsMajorTimeStep(Code MNLR v3 M);
tmp = false:
simulationData->mData->mIsSolverAssertCheck = tmp;
simulationData->mData->mIsSolverCheckingCIC = false;
tmp = rtsiIsSolverComputingJacobian(&Code_MNLR_v3_M->solverInfo);
simulationData->mData->mIsComputingJacobian = tmp;
simulationData->mData->mIsEvaluatingF0 = false;
simulationData->mData->mIsSolverRequestingReset = false;
simulationData->mData->mIsModeUpdateTimeStep = rtsiIsModeUpdateTimeStep
 (&Code MNLR v3 M->solverInfo);
tmp_1[0] = 0;
tmp 0[0] = Code MNLR v3 B.INPUT 12 1 1[0];
tmp_0[1] = Code_MNLR_v3_B.INPUT_12_1_1[1];
tmp_0[2] = Code_MNLR_v3_B.INPUT_12_1_1[2];
tmp_0[3] = Code_MNLR_v3_B.INPUT_12_1_1[3];
tmp_1[1] = 4;
tmp 0[4] = Code MNLR v3 B.INPUT 1 1 1[0];
tmp 0[5] = Code_MNLR_v3_B.INPUT_1_1_1[1];
tmp_0[6] = Code_MNLR_v3_B.INPUT_1_1_1[2];
tmp_0[7] = Code_MNLR_v3_B.INPUT_1_1_1[3];
tmp_1[2] = 8;
tmp_0[8] = Code_MNLR_v3_B.INPUT_13_1_1[0];
tmp_0[9] = Code_MNLR_v3_B.INPUT_13_1_1[1];
tmp_0[10] = Code_MNLR_v3_B.INPUT_13_1_1[2];
tmp 0[11] = Code MNLR v3 B.INPUT 13 1 1[3];
tmp 1[3] = 12;
tmp_0[12] = Code_MNLR_v3_B.INPUT_15_1_1[0];
tmp 0[13] = Code MNLR v3 B.INPUT 15 1 1[1];
tmp_0[14] = Code_MNLR_v3_B.INPUT_15_1_1[2];
tmp_0[15] = Code_MNLR_v3_B.INPUT_15_1_1[3];
tmp 1[4] = 16;
tmp_0[16] = Code_MNLR_v3_B.INPUT_21_1_1[0];
tmp 0[17] = Code MNLR v3 B.INPUT 21 1 1[1];
tmp_0[18] = Code_MNLR_v3_B.INPUT_21_1_1[2];
tmp_0[19] = Code_MNLR_v3_B.INPUT_21_1_1[3];
tmp 1[5] = 20;
tmp_0[20] = Code_MNLR_v3_B.INPUT_24_1_1[0];
tmp_0[21] = Code_MNLR_v3_B.INPUT_24_1_1[1];
tmp_0[22] = Code_MNLR_v3_B.INPUT_24_1_1[2];
tmp_0[23] = Code_MNLR_v3_B.INPUT_24_1_1[3];
tmp 1[6] = 24;
tmp_0[24] = Code_MNLR_v3_B.INPUT_18_1_1[0];
tmp_0[25] = Code_MNLR_v3_B.INPUT_18_1_1[1];
tmp 0[26] = Code MNLR v3 B.INPUT 18 1 1[2];
tmp_0[27] = Code_MNLR_v3_B.INPUT_18_1_1[3];
tmp 1[7] = 28;
tmp_0[28] = Code_MNLR_v3_B.INPUT_26_1_1[0];
```

tmp\_0[29] = Code\_MNLR\_v3\_B.INPUT\_26\_1\_1[1]; tmp 0[30] = Code MNLR v3 B.INPUT 26 1 1[2]; $tmp_0[31] = Code_MNLR_v3_B.INPUT_26_1_1[3];$  $tmp_1[8] = 32;$ tmp 0[32] = Code MNLR v3 B.INPUT 29 1 1[0];tmp\_0[33] = Code\_MNLR\_v3\_B.INPUT\_29\_1\_1[1]; tmp 0[34] = Code MNLR v3 B.INPUT 29 1 1[2];tmp\_0[35] = Code\_MNLR\_v3\_B.INPUT\_29\_1\_1[3];  $tmp_1[9] = 36;$  $tmp_0[36] = Code_MNLR_v3_B.INPUT_31_1_1[0];$  $tmp_0[37] = Code_MNLR_v3_B.INPUT_31_1_1[1];$ tmp 0[38] = Code MNLR v3 B.INPUT 31 1 1[2]; $tmp_0[39] = Code_MNLR_v3_B.INPUT_31_1_1[3];$  $tmp_1[10] = 40;$ tmp 0[40] = Code MNLR v3 B.INPUT 2 1 1[0]; $tmp_0[41] = Code_MNLR_v3_B.INPUT_2_1_1[1];$  $tmp_0[42] = Code_MNLR_v3_B.INPUT_2_1_1[2];$ tmp\_0[43] = Code\_MNLR\_v3\_B.INPUT\_2\_1\_1[3];  $tmp_1[11] = 44;$ tmp\_0[44] = Code\_MNLR\_v3\_B.INPUT\_28\_1\_1[0]; tmp\_0[45] = Code\_MNLR\_v3\_B.INPUT\_28\_1\_1[1];  $tmp_0[46] = Code_MNLR_v3_B.INPUT_28_1_1[2];$ tmp 0[47] = Code MNLR v3 B.INPUT 28 1 1[3]; $tmp_1[12] = 48;$  $tmp_0[48] = Code_MNLR_v3_B.INPUT_5_1_1[0];$  $tmp_0[49] = Code_MNLR_v3_B.INPUT_5_1_1[1];$ tmp\_0[50] = Code\_MNLR\_v3\_B.INPUT\_5\_1\_1[2]; tmp 0[51] = Code MNLR v3 B.INPUT 5 1 1[3]; $tmp_1[13] = 52;$  $tmp_0[52] = Code_MNLR_v3_B.INPUT_7_1_1[0];$ tmp\_0[53] = Code\_MNLR\_v3\_B.INPUT\_7\_1\_1[1];  $tmp_0[54] = Code_MNLR_v3_B.INPUT_7_1_1[2];$  $tmp_0[55] = Code_MNLR_v3_B.INPUT_7_1_1[3];$  $tmp_1[14] = 56;$ tmp\_0[56] = Code\_MNLR\_v3\_B.INPUT\_9\_1\_1[0]; tmp 0[57] = Code MNLR v3 B.INPUT 9 1 1[1]; $tmp_0[58] = Code_MNLR_v3_B.INPUT_9_1_1[2];$ tmp\_0[59] = Code\_MNLR\_v3\_B.INPUT\_9\_1\_1[3]; tmp 1[15] = 60; $tmp_0[60] = Code_MNLR_v3_B.INPUT_17_1_1[0];$  $tmp_0[61] = Code_MNLR_v3_B.INPUT_17_1_1[1];$  $tmp_0[62] = Code_MNLR_v3_B.INPUT_17_1_1[2];$ tmp\_0[63] = Code\_MNLR\_v3\_B.INPUT\_17\_1\_1[3]; tmp 1[16] = 64;tmp 0[64] = Code MNLR v3 B.INPUT 3 1 1[0]; $tmp_0[65] = Code_MNLR_v3_B.INPUT_3_1_1[1];$  $tmp_0[66] = Code_MNLR_v3_B.INPUT_3_1_1[2];$  $tmp_0[67] = Code_MNLR_v3_B.INPUT_3_1_1[3];$  $tmp_1[17] = 68;$  $tmp_0[68] = Code_MNLR_v3_B.INPUT_4_1_1[0];$ tmp\_0[69] = Code\_MNLR\_v3\_B.INPUT\_4\_1\_1[1]; tmp 0[70] = Code MNLR v3 B.INPUT 4 1 1[2]; $tmp_0[71] = Code_MNLR_v3_B.INPUT_4_1_1[3];$ tmp 1[18] = 72;tmp 0[72] = Code MNLR v3 B.INPUT 6 1 1[0];tmp\_0[73] = Code\_MNLR\_v3\_B.INPUT\_6\_1\_1[1];  $tmp_0[74] = Code_MNLR_v3_B.INPUT_6_1_1[2];$  $tmp_0[75] = Code_MNLR_v3_B.INPUT_6_1_1[3];$ 

 $tmp_1[19] = 76;$ tmp 0[76] = Code MNLR v3 B.INPUT 8 1 1[0];tmp\_0[77] = Code\_MNLR\_v3\_B.INPUT\_8\_1\_1[1];  $tmp_0[78] = Code_MNLR_v3_B.INPUT_8_1_1[2];$ tmp\_0[79] = Code\_MNLR\_v3\_B.INPUT\_8\_1\_1[3];  $tmp_1[20] = 80;$ tmp 0[80] = Code MNLR v3 B.INPUT 10 1 1[0]; $tmp_0[81] = Code_MNLR_v3_B.INPUT_10_1_1[1];$ tmp\_0[82] = Code\_MNLR\_v3\_B.INPUT\_10\_1\_1[2];  $tmp_0[83] = Code_MNLR_v3_B.INPUT_10_1_1[3];$  $tmp_1[21] = 84;$ tmp 0[84] = Code MNLR v3 B.INPUT 11 1 1[0];tmp 0[85] = Code MNLR v3 B.INPUT 11 1 1[1]; $tmp_0[86] = Code_MNLR_v3_B.INPUT_11_1[2];$  $tmp_0[87] = Code_MNLR_v3_B.INPUT_11_1[3];$  $tmp_1[22] = 88;$  $tmp_0[88] = Code_MNLR_v3_B.INPUT_14_1_1[0];$  $tmp_0[89] = Code_MNLR_v3_B.INPUT_14_1_1[1];$  $tmp_0[90] = Code_MNLR_v3_B.INPUT_14_1_1[2];$ tmp 0[91] = Code MNLR v3 B.INPUT 14 1 1[3];tmp 1[23] = 92; $tmp_0[92] = Code_MNLR_v3_B.INPUT_16_1_1[0];$ tmp 0[93] = Code MNLR v3 B.INPUT 16 1 1[1]; $tmp_0[94] = Code_MNLR_v3_B.INPUT_16_1_1[2];$  $tmp_0[95] = Code_MNLR_v3_B.INPUT_16_1_1[3];$ tmp 1[24] = 96;tmp\_0[96] = Code\_MNLR\_v3\_B.INPUT\_19\_1\_1[0]; tmp 0[97] = Code MNLR v3 B.INPUT 19 1 1[1];tmp 0[98] = Code\_MNLR\_v3\_B.INPUT\_19\_1\_1[2]; tmp\_0[99] = Code\_MNLR\_v3\_B.INPUT\_19\_1\_1[3]; tmp 1[25] = 100;tmp\_0[100] = Code\_MNLR\_v3\_B.INPUT\_20\_1\_1[0];  $tmp_0[101] = Code_MNLR_v3_B.INPUT_20_1_1[1];$ tmp 0[102] = Code MNLR v3 B.INPUT 20 1 1[2];tmp\_0[103] = Code\_MNLR\_v3\_B.INPUT\_20\_1\_1[3]; tmp 1[26] = 104;tmp\_0[104] = Code\_MNLR\_v3\_B.INPUT\_22\_1\_1[0]; tmp\_0[105] = Code\_MNLR\_v3\_B.INPUT\_22\_1\_1[1]; tmp 0[106] = Code MNLR v3 B.INPUT 22 1 1[2];tmp\_0[107] = Code\_MNLR\_v3\_B.INPUT\_22\_1\_1[3];  $tmp_1[27] = 108;$  $tmp_0[108] = Code_MNLR_v3_B.INPUT_25_1_1[0];$ tmp\_0[109] = Code\_MNLR\_v3\_B.INPUT\_25\_1\_1[1]; tmp 0[110] = Code MNLR v3 B.INPUT 25 1 1[2]; $tmp_0[111] = Code_MNLR_v3_B.INPUT_25_1_1[3];$  $tmp_1[28] = 112;$ tmp 0[112] = Code MNLR v3 B.INPUT 23 1 1[0];tmp\_0[113] = Code\_MNLR\_v3\_B.INPUT\_23\_1\_1[1];  $tmp_0[114] = Code_MNLR_v3_B.INPUT_23_1_1[2];$  $tmp_0[115] = Code_MNLR_v3_B.INPUT_23_1_1[3];$ tmp\_1[29] = 116; tmp 0[116] = Code MNLR v3 B.INPUT 27 1 1[0];tmp\_0[117] = Code\_MNLR\_v3\_B.INPUT\_27\_1\_1[1]; tmp\_0[118] = Code\_MNLR\_v3\_B.INPUT\_27\_1\_1[2]; tmp 0[119] = Code MNLR v3 B.INPUT 27 1 1[3]; $tmp_1[30] = 120;$ tmp\_0[120] = Code\_MNLR\_v3\_B.INPUT\_30\_1\_1[0];  $tmp_0[121] = Code_MNLR_v3_B.INPUT_30_1_1[1];$ 

```
tmp_0[122] = Code_MNLR_v3_B.INPUT_30_1_1[2];
 tmp_0[123] = Code_MNLR_v3_B.INPUT_30_1_1[3];
 tmp_1[31] = 124;
 tmp_0[124] = Code_MNLR_v3_B.INPUT_32_1_1[0];
 tmp_0[125] = Code_MNLR_v3_B.INPUT_32_1_1[1];
 tmp_0[126] = Code_MNLR_v3_B.INPUT_32_1_1[2];
 tmp_0[127] = Code_MNLR_v3_B.INPUT_32_1_1[3];
 tmp_1[32] = 128;
 tmp_0[128] = Code_MNLR_v3_B.INPUT_33_1_1[0];
 tmp_0[129] = Code_MNLR_v3_B.INPUT_33_1_1[1];
 tmp_0[130] = Code_MNLR_v3_B.INPUT_33_1_1[2];
 tmp 0[131] = Code MNLR v3 B.INPUT 33 1 1[3];
 tmp 1[33] = 132;
 simulationData->mData->mInputValues.mN = 132;
 simulationData->mData->mInputValues.mX = &tmp 0[0];
 simulationData->mData->mInputOffsets.mN = 34;
 simulationData->mData->mInputOffsets.mX = &tmp_1[0];
 tmp 2 = \text{Code MNLR v3 M->massMatrixPr};
 tmp_3 = double_pointer_shift(tmp_2, Code_MNLR_v3_DW.STATE_1_MASS_MATRIX_PR);
 simulationData->mData->mMassMatrixPr.mN = 113;
 simulationData->mData->mMassMatrixPr.mX = tmp 3;
 diagnosticManager = (NeuDiagnosticManager *)Code_MNLR_v3_DW.STATE_1_DiagMgr;
 diagnosticTree = neu diagnostic manager get initial tree(diagnosticManager);
 tmp_4 = ne_simulator_method((NeslSimulator *)Code_MNLR_v3_DW.STATE_1_Simulator,
  NESL_SIM_MASSMATRIX, simulationData, diagnosticManager);
 if (tmp 4!=0) {
  tmp = error_buffer_is_empty(rtmGetErrorStatus(Code_MNLR_v3_M));
  if (tmp) {
   msg = rtw_diagnostics_msg(diagnosticTree);
   rtmSetErrorStatus(Code_MNLR_v3_M, msg);
  ł
 /* End of MassMatrix for SimscapeExecutionBlock: '<S954>/STATE_1' */
}
void local_evaluateMassMatrix(RTWSolverInfo *si, real_T *Mdest )
{
 /* Refresh global mass matrix */
 Code_MNLR_v3_massmatrix();
 /* Copy the mass matrix from system to the destination, if needed. */
 if (Mdest != rtsiGetSolverMassMatrixPr(si)) {
  real T *Msrc = rtsiGetSolverMassMatrixPr(si);
  int_T nzmax = rtsiGetSolverMassMatrixNzMax(si);
  (void) memcpy(Mdest, Msrc,
         (uint T)nzmax*sizeof(real T));
 }
}
void local evaluateFminusMv(RTWSolverInfo *si, const real T *v, real T *fminusMv
)
{
 /* Refresh forcing function */
 rtsiSetdX(si,fminusMv);
 Code MNLR v3 forcingfunction();
 /* Refresh global mass matrix */
 Code MNLR v3 massmatrix();
 /* Form f - M*v */
 ł
  real_T *elptr = rtsiGetSolverMassMatrixPr(si);
```

```
int_T *iptr = rtsiGetSolverMassMatrixIr(si);
  int_T *jc = rtsiGetSolverMassMatrixJc(si);
  int_T nx = 281;
  int_T col,row;
  for (col = 0; col < nx; col++) {
   for (row = jc[col]; row < jc[col+1]; row++) {
     fminusMv[*iptr++] -= (*v) * (*elptr++);
   }
   v++;
  }
 }
}
/* Simplified version of numjac.cpp, for use with RTW. */
void local_numjac( RTWSolverInfo *si, real_T *y, const real_T *v, const real_T
           *Fty, real_T *fac, real_T *dFdy )
{
 /* constants */
 real_T THRESH = 1e-6;
 real_T EPS = 2.2e-16;
                                 /* utGetEps(); */
 real T BL = pow(EPS, 0.75);
 real_T BU = pow(EPS, 0.25);
 real T FACMIN = pow(EPS, 0.78);
 real_T FACMAX = 0.1;
 int_T nx = 281;
 real_T *x = rtsiGetContStates(si);
 real_T del;
 real T difmax;
 real T FdelRowmax;
 real_T temp;
 real T Fdiff;
 real_T maybe;
 real_T xscale;
 real T fscale;
 real_T *p;
 int T rowmax;
 int_T i,j;
 if (x \mid = y)
  (void) memcpy(x, y, y)
           (uint_T)nx*sizeof(real_T));
 rtsiSetSolverComputingJacobian(si,true);
 for (p = dFdy, j = 0; j < nx; j++, p += nx) {
  /* Select an increment del for a difference approximation to
    column j of dFdy. The vector fac accounts for experience
    gained in previous calls to numjac. */
  xscale = fabs(x[j]);
  if (xscale < THRESH)
   xscale = THRESH;
  temp = (x[j] + fac[j]*xscale);
  del = temp - y[j];
  while (del == 0.0) {
   if (fac[j] < FACMAX) {
     fac[j] *= 100.0;
    if (fac[j] > FACMAX)
      fac[j] = FACMAX;
    temp = (x[j] + fac[j]*xscale);
     del = temp - x[j];
    } else {
```

```
/* thresh is nonzero */
    del = THRESH;
    break;
   }
  }
  /* Keep del pointing into region. */
  if (Fty[j] \ge 0.0)
   del = fabs(del);
  else
   del = -fabs(del);
  /* Form a difference approximation to column j of dFdy. */
  temp = x[j];
  x[i] += del;
  Code_MNLR_v3_step();
  local_evaluateFminusMv(si,v,p );
  x[i] = temp;
  difmax = 0.0;
  rowmax = 0;
  FdelRowmax = p[0];
  temp = 1.0 / \text{del};
  for (i = 0; i < nx; i++) {
   Fdiff = p[i] - Fty[i];
   maybe = fabs(Fdiff);
   if (maybe > difmax) {
    difmax = maybe;
    rowmax = i;
    FdelRowmax = p[i];
   }
   p[i] = temp * Fdiff;
  }
  /* Adjust fac for next call to numjac. */
  if (((FdelRowmax != 0.0) && (Fty[rowmax] != 0.0)) || (difmax == 0.0)) {
   fscale = fabs(FdelRowmax);
   if (fscale < fabs(Fty[rowmax]))
    fscale = fabs(Fty[rowmax]);
   if (difmax <= BL*fscale) {
    /* The difference is small, so increase the increment. */
    fac[j] *= 10.0;
    if (fac[j] > FACMAX)
      fac[j] = FACMAX;
   } else if (difmax > BU*fscale) {
    /* The difference is large, so reduce the increment. */
    fac[j] *= 0.1;
    if (fac[j] < FACMIN)
      fac[j] = FACMIN;
   }
  }
 }
 rtsiSetSolverComputingJacobian(si,false);
                        /* end local_numjac */
}
/*
* This function updates continuous states using the ODE14X fixed-step
* solver algorithm
*/
static void rt ertODEUpdateContinuousStates(RTWSolverInfo *si)
 /* Solver Matrices */
 static int_T rt_ODE14x_N[4] = { 12, 8, 6, 4 };
```

{

```
time_T t0 = rtsiGetT(si);
time T t1 = t0;
time_T h = rtsiGetStepSize(si);
real_T *x1 = rtsiGetContStates(si);
int T order = rtsiGetSolverExtrapolationOrder(si);
int_T numIter = rtsiGetSolverNumberNewtonIterations(si);
ODE14X IntgData *id = (ODE14X IntgData *)rtsiGetSolverData(si);
real_T *x0 = id > x0;
real_T *f0 = id > f0;
real_T *x1start = id->x1start;
real_T *f1 = id->f1;
real_T *Delta = id->Delta;
real_T *E = id ->E;
real_T *fac = id->fac;
real_T *dfdx = id->DFDX;
real_T *W = id - W;
int_T *pivots = id->pivots;
real T *xtmp = id->xtmp;
real_T *ztmp = id->ztmp;
int T *Mpattern ir = rtsiGetSolverMassMatrixIr(si);
int_T *Mpattern_jc = rtsiGetSolverMassMatrixJc(si);
real_T *M = id > M;
real T * M1 = id \rightarrow M1;
real_T *xdot = id->xdot;
real_T *Edot = id->Edot;
real_T *fminusMxdot = id->fminusMxdot;
int_T col,row,rowidx;
int T * N = \&(rt ODE14x N[0]);
int_T i,j,k,iter;
int_T nx = 281;
rtsiSetSimTimeStep(si,MINOR TIME STEP);
/* Save the state values at time t in y, we'll use x as ynew. */
(void) memcpy(x0, x1,
        (uint T)nx*sizeof(real T));
/* Assumes that rtsiSetT and ModelOutputs are up-to-date */
local evaluateMassMatrix(si,M);
rtsiSetdX(si, xdot);
Code_MNLR_v3_derivatives();
/* f0 = f(t,y) */
rtsiSetdX(si, f0);
Code_MNLR_v3_forcingfunction();
/* Form fminusMxdot = f(x) - M(x)*xdot, d(fminusMxdot)/dx = df/dx - d(Mv)/dx */
(void) memcpy(fminusMxdot, f0,
        (uint T)nx*sizeof(real T));
for (col = 0; col < nx; col++) {
 for (rowidx = Mpattern_jc[col]; rowidx < Mpattern_jc[col+1]; rowidx++) {
  real T m row col = M[rowidx];
  row = Mpattern_ir[rowidx];
  fminusMxdot[row] -= m_row_col*xdot[col];
 }
}
local numjac(si,x0,xdot,fminusMxdot,fac,dfdx );
for (j = 0; j < order; j++) {
 real_T *p;
 real T hN = h/N[j];
 /* Get the iteration matrix and solution at t0 */
 /* [L,U] = lu(M - hN*J) */
 (void) memcpy(W, dfdx,
```

```
(uint_T)nx*nx*sizeof(real_T));
for (p = W, i = 0; i < nx*nx; i++, p++) {
 *p *= (-hN);
for (col = 0, p = W; col < nx; col++, p += nx) {
 for (rowidx = Mpattern_jc[col]; rowidx < Mpattern_jc[col+1]; rowidx++) {
  real T m row col = M[rowidx];
  row = Mpattern_ir[rowidx];
  p[row] += m_row_col;
 }
}
rt lu real(W, nx,
       pivots):
/* First Newton's iteration at t0. */
/* rhs = hN*f0 */
for (i = 0; i < nx; i++) {
 Delta[i] = hN*f0[i];
}
/* Delta = (U \ (L \ rhs)) */
rt ForwardSubstitutionRR Dbl(W, Delta,
 f1. nx.
 1, pivots,
 1);
rt_BackwardSubstitutionRR_Dbl(W+nx*nx-1, f1+nx-1,
 Delta, nx,
 1, 0);
/* ytmp = y0 + Delta
 ztmp = (ytmp-y0)/h
*/
(void) memcpy(x1, x0,
        (uint_T)nx*sizeof(real_T));
for (i = 0; i < nx; i++) {
 x1[i] += Delta[i];
 ztmp[i] = Delta[i]/hN;
}
/* Additional Newton's iterations, if desired.
  for iter = 2:NewtIter
  rhs = hN*feval(odefun,tn,ytmp,extraArgs{:}) - M*(ytmp - yn);
  if statedepM % only for state dep. Mdel \sim = 0
  Mdel = M - feval(massfun,tn,ytmp);
  rhs = rhs + Mdel*ztmp*h;
  end
  Delta = (U \setminus (L \setminus rhs));
  ytmp = ytmp + Delta;
  ztmp = (ytmp - yn)/h
 end
*/
rtsiSetT(si, t0);
rtsiSetdX(si, f1);
for (iter = 1; iter < numIter; iter++) {
 Code_MNLR_v3_step();
 Code MNLR v3 forcingfunction();
 for (i = 0; i < nx; i++) {
  Delta[i] = hN*f1[i];
  xtmp[i] = x1[i] - x0[i];
 }
 /* rhs = hN*f(tn,ytmp) - M*(ytmp-yn) */
 for (col = 0; col < nx; col++) {
```

```
for (rowidx = Mpattern_jc[col]; rowidx < Mpattern_jc[col+1]; rowidx++) {
   real_T m_row_col = M[rowidx];
   row = Mpattern_ir[rowidx];
   Delta[row] -= m_row_col*xtmp[col];
  }
 }
 /* rhs = rhs - (Mtmp - M)*ztmp*h */
 local_evaluateMassMatrix(si,M1 );
 for (i = 0; i < rtsiGetSolverMassMatrixNzMax(si); i++) {
  M1[i] = M[i];
 ł
 for (col = 0; col < nx; col++) {
  for (rowidx = Mpattern jc[col]; rowidx < Mpattern jc[col+1]; rowidx++) {
   real_T m_row_col = M1[rowidx];
   row = Mpattern ir[rowidx];
   Delta[row] -= hN*m_row_col*ztmp[col];
  }
 }
 rt_ForwardSubstitutionRR_Dbl(W, Delta,
  f1. nx.
  1, pivots,
  1);
 rt BackwardSubstitutionRR Dbl(W+nx*nx-1, f1+nx-1,
  Delta, nx,
  1, 0);
 /* ytmp = ytmp + delta
   ztmp = (ytmp - yn)/h
  */
 for (i = 0; i < nx; i++) {
  x1[i] += Delta[i];
  ztmp[i] = (x1[i] - x0[i])/hN;
 }
/* Steps from t0+hN to t1 -- subintegration of N(j) steps for extrapolation
  ttmp = t0;
  for i = 2:N(j)
  ttmp = ttmp + hN
  ytmp0 = ytmp;
  for iter = 1:NewtIter
  rhs = (ytmp0 - ytmp) + hN*feval(odefun,ttmp,ytmp,extraArgs{:});
  Delta = (U \setminus (L \setminus rhs));
  ytmp = ytmp + Delta;
  end
 end
 */
for (k = 1; k < N[j]; k++) {
 t1 = t0 + k*hN;
 (void) memcpy(x1start, x1,
          (uint_T)nx*sizeof(real_T));
 rtsiSetT(si, t1);
 rtsiSetdX(si, f1);
 for (iter = 0; iter < numIter; iter++) {
  Code_MNLR_v3_step();
  Code_MNLR_v3_forcingfunction();
  if (iter == 0) {
   for (i = 0; i < nx; i++) {
     Delta[i] = hN*f1[i];
```

}

}

```
} else {
   for (i = 0; i < nx; i++) {
    Delta[i] = hN*f1[i];
    xtmp[i] = (x1[i]-x1start[i]);
   }
   /* rhs = hN*f(tn,ytmp) - M*(ytmp-yn) */
   for (col = 0; col < nx; col++) {
    for (rowidx = Mpattern_jc[col]; rowidx < Mpattern_jc[col+1]; rowidx
       ++) {
      real_T m_row_col = M[rowidx];
     row = Mpattern_ir[rowidx];
     Delta[row] -= m row col*xtmp[col];
     }
   }
  }
  /* For state-dep., Mdel = M(ttmp,ytmp) - M */
  Code_MNLR_v3_step();
  local evaluateMassMatrix(si,M1);
  for (i = 0; i < rtsiGetSolverMassMatrixNzMax(si); i++) {
   M1[i] = M[i];
  }
  /* rhs = rhs - Mdel*ztmp*h */
  for (col = 0; col < nx; col++) {
   for (rowidx = Mpattern_jc[col]; rowidx < Mpattern_jc[col+1]; rowidx++)
   {
    real_T m_row_col = M1[rowidx];
    row = Mpattern_ir[rowidx];
    Delta[row] -= hN*m row col*ztmp[col];
   }
  }
  rt ForwardSubstitutionRR Dbl(W, Delta,
   f1, nx,
   1, pivots,
   1);
  rt_BackwardSubstitutionRR_Dbl(W+nx*nx-1, f1+nx-1,
   Delta, nx,
   1, 0);
  /* ytmp = ytmp + Delta
    ztmp = (ytmp - ytmp0)/h
   */
  for (i = 0; i < nx; i++)
   x1[i] += Delta[i];
   ztmp[i] = (x1[i] - x1start[i])/hN;
  }
 }
}
/* Extrapolate to order j
 E(:,j) = ytmp
 for k = j:-1:2
 coef = N(k-1)/(N(j) - N(k-1))
 E(:,k-1) = E(:,k) + coef^{*}(E(:,k) - E(:,k-1))
 end
*/
(void) memcpy(&(E[nx*j]), x1,
        (uint_T)nx*sizeof(real_T));
for (k = j; k > 0; k--) {
 real_T coef = (real_T)(N[k-1]) / (N[j]-N[k-1]);
 for (i = 0; i < nx; i++) {
```

```
x1[i] = E[nx^{k+i}] + coef^{k}(E[nx^{k+i}] - E[nx^{k+i}]);
   }
   (void) memcpy(\&(E[nx^{*}(k-1)]), x1,
            (uint_T)nx*sizeof(real_T));
  /* Extrapolate the derivative */
  for (i = 0; i < nx; i++) {
   xdot[i] = (x1[i] - x1start[i])/hN;
  }
  (void) memcpy(&(Edot[nx*j]), xdot,
           (uint_T)nx*sizeof(real_T));
  for (k = j; k > 0; k--) {
   real_T coef = (real_T)(N[k-1]) / (N[j]-N[k-1]);
   for (i = 0; i < nx; i++) {
     xdot[i] = Edot[nx*k+i] + coef*(Edot[nx*k+i] - Edot[nx*(k-1)+i]);
   ł
   (void) memcpy(&(Edot[nx*(k-1)]), xdot,
            nx*sizeof(real_T));
  }
 }
 /* x1 = E(:,1); */
 (void) memcpy(x1, E,
          (uint T)nx*sizeof(real T));
 /* Extrapolated xdot */
 (void) memcpy(xdot, Edot,
          nx*sizeof(real T));
 /* t1 = t0 + h; */
 rtsiSetT(si,rtsiGetSolverStopTime(si));
 rtsiSetSimTimeStep(si,MAJOR_TIME_STEP);
/*
* Output and update for atomic system:
*
    '<S17>/Evaluate Rule Consequents'
*
    '<S20>/Evaluate Rule Consequents'
*/
void Code MN EvaluateRuleConsequents(const real T rtu antecedentOutputs[25],
 real_T rty_aggregatedOutputs[4])
 real T outputMFCache[20];
 int32_T ruleID;
 static const int8_T b[100] = { 3, 1, 2, 3, 3, 1, 2, 3, 3, 5, 5, 5, 4, 3, 3, 3,
  3, 3, 3, 3, 3, 1, 1, 1, 2, 3, 3, 3, 2, 1, 1, 2, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1, 2,
  5, 5, 5, 4, 3, 3, 3, 3, 3, 1, 2, 3, 3, 3, 3, 2, 1, 3, 3, 3, 3, 5, 5, 5, 4, 1,
  1, 1, 2, 3, 3, 3, 3, 3, 3, 3, 2, 1, 3, 3, 2, 1, 1, 1, 1, 2, 3, 3, 3, 3, 3, 3, 3,
  3, 3, 5, 5, 5, 4;
 rty_aggregatedOutputs[0] = 0.0;
 rty aggregatedOutputs[1] = 0.0;
 rty_aggregatedOutputs[2] = 0.0;
 rty_aggregatedOutputs[3] = 0.0;
 outputMFCache[0] = -0.25;
 outputMFCache[1] = -0.125;
 outputMFCache[2] = 0.0;
 outputMFCache[3] = 0.125;
 outputMFCache[4] = 0.25;
 outputMFCache[5] = -0.25;
 outputMFCache[6] = -0.125;
 outputMFCache[7] = 0.0;
 outputMFCache[8] = 0.125;
```

ł

{

```
outputMFCache[9] = 0.25;
 outputMFCache[10] = -0.25;
 outputMFCache[11] = -0.125;
 outputMFCache[12] = 0.0;
 outputMFCache[13] = 0.125;
 outputMFCache[14] = 0.25;
 outputMFCache[15] = -0.25;
 outputMFCache[16] = -0.125;
 outputMFCache[17] = 0.0;
 outputMFCache[18] = 0.125;
 outputMFCache[19] = 0.25;
 for (ruleID = 0; ruleID < 25; ruleID++) {
  rty aggregatedOutputs[0] += outputMFCache[b[ruleID] - 1] *
   rtu_antecedentOutputs[ruleID];
  rty aggregatedOutputs[1] += outputMFCache[b[ruleID + 25] + 4] *
   rtu_antecedentOutputs[ruleID];
  rty_aggregatedOutputs[2] += outputMFCache[b[ruleID + 50] + 9] *
   rtu antecedentOutputs[ruleID];
  rty_aggregatedOutputs[3] += outputMFCache[b[ruleID + 75] + 14] *
   rtu antecedentOutputs[ruleID];
 }
ł
/*
* Output and update for atomic system:
   '<S18>/Evaluate Rule Consequents'
*
    '<S21>/Evaluate Rule Consequents'
*/
void Code EvaluateRuleConsequents p(real T rtu inputs, real T rtu inputs b,
 const real_T rtu_antecedentOutputs[25], real_T rty_aggregatedOutputs[4],
 B_EvaluateRuleConsequents_C_f_T *localB)
ł
 real_T outputMFCache[20];
 int32 T ruleID;
 static const int8_T b[100] = { 3, 1, 2, 3, 3, 1, 2, 3, 3, 5, 5, 5, 4, 3, 3, 3,
  3, 3, 3, 3, 3, 1, 1, 1, 2, 3, 3, 3, 2, 1, 1, 2, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1, 2,
  5, 5, 5, 4, 3, 3, 3, 3, 3, 1, 2, 3, 3, 3, 3, 2, 1, 3, 3, 3, 3, 5, 5, 5, 4, 1,
  1, 1, 2, 3, 3, 3, 3, 3, 3, 3, 2, 1, 3, 3, 2, 1, 1, 1, 1, 2, 3, 3, 3, 3, 3, 3, 3,
  3, 3, 5, 5, 5, 4;
 /* SignalConversion generated from: '<S125>/ SFunction ' */
 localB->TmpSignalConversionAtSFunctionI[0] = rtu_inputs;
 localB->TmpSignalConversionAtSFunctionI[1] = rtu_inputs_b;
 rty_aggregatedOutputs[0] = 0.0;
 rty aggregatedOutputs[1] = 0.0;
 rty_aggregatedOutputs[2] = 0.0;
 rty_aggregatedOutputs[3] = 0.0;
 outputMFCache[0] = -0.25;
 outputMFCache[1] = -0.125;
 outputMFCache[2] = 0.0;
 outputMFCache[3] = 0.125;
 outputMFCache[4] = 0.25;
 outputMFCache[5] = -0.25;
 outputMFCache[6] = -0.125;
 outputMFCache[7] = 0.0;
 outputMFCache[8] = 0.125;
 outputMFCache[9] = 0.25;
 outputMFCache[10] = -0.25;
 outputMFCache[11] = -0.125;
```

```
outputMFCache[12] = 0.0;
 outputMFCache[13] = 0.125;
 outputMFCache[14] = 0.25;
 outputMFCache[15] = -0.25;
 outputMFCache[16] = -0.125;
 outputMFCache[17] = 0.0;
 outputMFCache[18] = 0.125;
 outputMFCache[19] = 0.25;
 for (ruleID = 0; ruleID < 25; ruleID++) {
  rty_aggregatedOutputs[0] += outputMFCache[b[ruleID] - 1] *
   rtu_antecedentOutputs[ruleID];
  rty aggregatedOutputs[1] += outputMFCache[b[ruleID + 25] + 4] *
   rtu antecedentOutputs[ruleID];
  rty_aggregatedOutputs[2] += outputMFCache[b[ruleID + 50] + 9] *
   rtu antecedentOutputs[ruleID];
  rty_aggregatedOutputs[3] += outputMFCache[b[ruleID + 75] + 14] *
   rtu_antecedentOutputs[ruleID];
 }
}
/* Function for MATLAB Function: '<S17>/Evaluate Rule Antecedents' */
static real T Code MNLR v3 trimf(real T x, const real T params[3])
 real Ty;
 y = 0.0;
 if ((params[0] != params[1]) && (params[0] < x) && (x < params[1])) {
  y = 1.0 / (params[1] - params[0]) * (x - params[0]);
 }
 if ((params[1] != params[2]) \&\& (params[1] < x) \&\& (x < params[2])) 
  y = 1.0 / (params[2] - params[1]) * (params[2] - x);
 if (x = params[1]) {
  y = 1.0;
 }
 return y;
/* Model step function */
void Code_MNLR_v3_step(void)
ł
 if (rtmIsMajorTimeStep(Code MNLR v3 M)) {
  /* set solver stop time */
  rtsiSetSolverStopTime(&Code_MNLR_v3_M->solverInfo,
               ((Code_MNLR_v3_M->Timing.clockTick0+1)*
   Code_MNLR_v3_M->Timing.stepSize0));
                       /* end MajorTimeStep */
 }
 /* Update absolute time of base rate at minor time step */
 if (rtmIsMinorTimeStep(Code_MNLR_v3_M)) {
  Code MNLR v3 M->Timing.t[0] = rtsiGetT(&Code MNLR v3 M->solverInfo);
 ł
 ł
  NeParameterBundle expl_temp;
  NeslSimulationData *simulationData;
  NeuDiagnosticManager *diag;
  NeuDiagnosticTree *diagTree;
  NeuDiagnosticTree *diagnosticTree;
  NeuDiagnosticTree *diagnosticTree 0;
  char *msg;
  char *msg_0;
  char *msg_1;
```

real\_T tmp\_1[132]; real T rtb antecedentOutputs b[29]; real\_T rtb\_antecedentOutputs\_c[25]; real\_T inputMFCache\_0[23]; real\_T outputMFCache[20]; real\_T tmp[16]; real T inputMFCache[10]; real\_T rtb\_aggregatedOutputs\_a[4]; real\_T tmp\_5[3]; real\_T InputConversion\_o\_idx\_0; real\_T InputConversion\_o\_idx\_1; real T InputConversion o idx 2; real T Sum2; real\_T Sum22; real T Sum26; real\_T Sum34; real\_T Sum35; real T Sum5; real\_T Sum6; real T Sum fa; real T Sum fa tmp; real\_T TmpSignalConversionAtSFunctio\_k; real TY; real\_T Y\_f; real\_T mVal; real\_T rtb\_defuzzifiedOutputs\_g\_idx\_0; real\_T rtb\_defuzzifiedOutputs\_g\_idx\_3; real T rtb defuzzifiedOutputs o idx 0; real T rtb defuzzifiedOutputs o idx 1; real\_T rtb\_defuzzifiedOutputs\_o\_idx\_2; real T rtb defuzzifiedOutputs o idx 3; real\_T rtb\_defuzzifiedOutputs\_ol; real\_T time; real T time 0; real\_T time\_1; real T time 2; real\_T \*parameterBundle\_mRealParameters; int32 T Sum9; int32 T mfIndex; int32\_T ruleID; int32\_T tmp\_6; int32\_T tmp\_7; int32\_T tmp\_8; int T tmp 4[35]; int\_T tmp\_2[34]; int8\_T b; boolean T ok; boolean\_T tmp\_0; boolean\_T tmp\_3; static const real\_T c[3] = { -4.5, -0.1905, 4.5 }; static const int8\_T b\_0[50] = { 3, 1, 2, 4, 5, 3, 3, 3, 3, 5, 5, 4, 4, 5, 5, 4, 4, 1, 1, 2, 2, 1, 1, 2, 2, 3, 3, 3, 3, 3, 3, 1, 2, 4, 5, 5, 4, 5, 4, 1, 2, 1, 2, 5, 4, 5, 4, 1, 2, 1, 2; static const real\_T i[3] = { -0.125, -1.11E-18, 0.125 }; static const real T k[3] = { 0.06, 0.06907, 0.2491 }; static const real\_T  $g[3] = \{ 0.15, 0.25, 0.35 \};$ static const real\_T  $j[3] = \{ 0.25, 0.43, 0.44 \};$ static const real\_T  $e[3] = \{ -10.0, 0.06, 0.06 \};$ 

static const real\_T d[3] = { 0.44, 0.44, 10.0 }; static const real\_T h[3] = { 0.06, 0.07, 0.25 }; static const real\_T  $f[3] = \{ 0.249, 0.429, 0.44 \};$ static const int8\_T b\_1[116] = { 3, 3, 3, 3, 3, 3, 3, 3, 2, 4, 4, 3, 2, 3, 3, 4, 3, 2, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 4, 2, 3, 3, 3, 3, 3, 3, 3, 3, 2, 4, 3, 2, 3, 4, 3, 4, 3, 2, 3, 3, 2, 4, 3, 2, 3, 3, 3, 3, 3, 4, 2, 3, 3, 3, 3, 3, 3, 3, 3, 2, 4, 3, 2, 3, 4, 3, 4, 3, 2, 2, 4, 3, 3, 2, 3, 3, 3, 3, 3, 3, 2, 4, 3, 3, 3, 3, 3, 3, 3, 2, 4, 3, 2, 3, 4, 2, 3, 4, 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, 3, 2, 4 }; static const int8\_T b\_2[145] = { 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 3, 3, 3, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 7, 6, 6, 7, 0, 0, 4, 2, 0, 0, 0, 0, -6, 6, -7, 7, 0, 0, 0, 0, 0, -6, 7, 2, 4, 3, 3, -6, -7, 3, 6, 7, 0, 0, 6, 7, 4, 2, -7, 7, -6, 6, 0, 0, 0, 0, 0, 0, 0, 0, 7, -6, 3, 3, 2, 4, -7, -6 }; static const int8\_T c\_0[5] = { 0, 3, 6, 9, 16 }; if (rtmIsMajorTimeStep(Code MNLR v3 M)) { /\* SimscapeRtp: '<S952>/RTP\_1' incorporates: Constant: '<Root>/Subsystem\_around\_RTP\_0D87B3D8\_x' \* Constant: '<Root>/Subsystem\_around\_RTP\_107624C3\_x' Constant: '<Root>/Subsystem\_around\_RTP\_14ABBE51\_x' \* \* Constant: '<Root>/Subsystem around RTP 1F6FC6C6 x' \* Constant: '<Root>/Subsystem\_around\_RTP\_27882079\_x' \* Constant: '<Root>/Subsystem\_around\_RTP\_508F10EF\_x' \* Constant: '<Root>/Subsystem\_around\_RTP\_649653A0\_x' \* Constant: '<Root>/Subsystem\_around\_RTP\_64C14ADE\_x' \* Constant: '<Root>/Subsystem\_around\_RTP\_6868F650\_x' \* Constant: '<Root>/Subsystem around RTP 8DA2EFEB x' \* Constant: '<Root>/Subsystem\_around\_RTP\_8FD0DB57\_x' \* Constant: '<Root>/Subsystem\_around\_RTP\_BE8171C3\_x' \* Constant: '<Root>/Subsystem\_around\_RTP\_C9864155\_x' Constant: '<Root>/Subsystem\_around\_RTP\_E542101F\_x' \* \* Constant: '<Root>/Subsystem around RTP E6A2A8B4 x' Constant: '<Root>/Subsystem\_around\_RTP\_FAA5DF7D\_x' \*/ if (Code\_MNLR\_v3\_DW.RTP\_1\_SetParametersNeeded) { tmp[0] = 20.0;tmp[1] = 20.0;tmp[2] = 20.0;tmp[3] = 20.0;tmp[4] = 20.0;tmp[5] = 20.0;tmp[6] = 20.0;tmp[7] = 20.0;tmp[8] = 20.0;tmp[9] = 20.0;tmp[10] = 20.0;tmp[11] = 20.0;tmp[12] = 20.0;tmp[13] = 20.0;tmp[14] = 20.0;tmp[15] = 20.0;parameterBundle mRealParameters = &tmp[0]; diag = rtw create diagnostics(); diagTree = neu\_diagnostic\_manager\_get\_initial\_tree(diag); expl\_temp.mRealParameters.mN = 16; expl\_temp.mRealParameters.mX = parameterBundle\_mRealParameters;

```
expl_temp.mLogicalParameters.mN = 0;
  expl_temp.mLogicalParameters.mX = NULL;
  expl_temp.mIntegerParameters.mN = 0;
  expl_temp.mIntegerParameters.mX = NULL;
  expl_temp.mIndexParameters.mN = 0;
  expl_temp.mIndexParameters.mX = NULL;
  ok = nesl rtp manager set rtps((NeslRtpManager *)
   Code_MNLR_v3_DW.RTP_1_RtpManager, Code_MNLR_v3_M->Timing.t[0],
   expl_temp, diag);
  if (!ok) {
   ok = error_buffer_is_empty(rtmGetErrorStatus(Code_MNLR_v3_M));
   if (ok) {
    msg = rtw diagnostics msg(diagTree);
    rtmSetErrorStatus(Code_MNLR_v3_M, msg);
   }
 }
 Code MNLR v3 DW.RTP 1 SetParametersNeeded = false;
/* End of SimscapeRtp: '<S952>/RTP_1' */
}
/* Step: '<Root>/Step13' incorporates:
* Sin: '<Root>/Sine Wave2'
* Step: '<Root>/Step1'
* Step: '<Root>/Step10'
* Step: '<Root>/Step11'
* Step: '<Root>/Step12'
* Step: '<Root>/Step14'
* Step: '<Root>/Step15'
* Step: '<Root>/Step17'
* Step: '<Root>/Step18'
* Step: '<Root>/Step19'
*
  Step: '<Root>/Step2'
* Step: '<Root>/Step20'
* Step: '<Root>/Step22'
* Step: '<Root>/Step23'
* Step: '<Root>/Step24'
  Step: '<Root>/Step25'
*
  Step: '<Root>/Step3'
*
  Step: '<Root>/Step30'
* Step: '<Root>/Step31'
* Step: '<Root>/Step32'
* Step: '<Root>/Step33'
*
  Step: '<Root>/Step34'
* Step: '<Root>/Step35'
* Step: '<Root>/Step4'
* Step: '<Root>/Step5'
* Step: '<Root>/Step7'
*
  Step: '<Root>/Step8'
*
  Step: '<Root>/Step9'
* Step: '<S103>/Step1'
* Step: '<S54>/Step'
* Step: '<S55>/Step'
* Step: '<S63>/Step'
* Step: '<S64>/Step'
* Step: '<S65>/Step'
* Step: '<S66>/Step'
```

\* Step: '<S68>/Step'

```
* Step: '<S69>/Step'
*/
Sum_fa_tmp = Code_MNLR_v3_M->Timing.t[0];
if (Sum_fa_tmp < 2.5) {
 Sum 26 = 0.0;
} else {
 Sum 26 = -4.5;
}
/* Step: '<Root>/Step12' */
if (Sum_fa_tmp < 3.0) {
Sum22 = 0.0;
} else {
 Sum22 = 4.5;
/* Sum: '<Root>/Sum8' incorporates:
* Step: '<Root>/Step12'
* Step: '<Root>/Step13'
* Sum: '<Root>/Sum19'
*/
Y = Sum26 + Sum22;
/* Step: '<Root>/Step4' */
if (Sum_fa_tmp < 3.5) {
ruleID = 0;
} else {
 ruleID = -5;
ł
/* Step: '<Root>/Step1' */
if (Sum fa tmp < 4.0) {
 Sum9 = 0;
} else {
 Sum9 = 5;
ł
/* Sum: '<Root>/Sum9' incorporates:
* Step: '<Root>/Step1'
* Step: '<Root>/Step4'
*/
Sum9 += ruleID;
/* Outputs for Atomic SubSystem: '<Root>/Fuzzy Logic Controller2' */
/* MATLAB Function: '<S18>/Evaluate Rule Antecedents' incorporates:
* SignalConversion generated from: '<S124>/ SFunction '
*/
Sum 26 = 0.0;
tmp_5[0] = -13.5;
tmp 5[1] = -9.0;
tmp_5[2] = -4.5;
inputMFCache[0] = Code_MNLR_v3_trimf(Y, tmp_5);
tmp_5[0] = -9.0;
tmp_5[1] = -4.5;
tmp_5[2] = 0.0;
inputMFCache[1] = Code_MNLR_v3_trimf(Y, tmp_5);
inputMFCache[2] = Code_MNLR_v3_trimf(Y, c);
tmp 5[0] = 0.0;
tmp_5[1] = 4.5;
tmp_5[2] = 9.0;
inputMFCache[3] = Code_MNLR_v3_trimf(Y, tmp_5);
tmp_5[0] = 4.5;
tmp_5[1] = 9.0;
tmp_5[2] = 13.5;
```

inputMFCache[4] = Code\_MNLR\_v3\_trimf(Y, tmp\_5); tmp 5[0] = -9.0; $tmp_5[1] = -6.0;$  $tmp_5[2] = -3.0;$ inputMFCache[5] = Code MNLR v3 trimf((real T)Sum9, tmp 5);  $tmp_5[0] = -6.0;$ tmp 5[1] = -3.0; $tmp_5[2] = 0.0;$ inputMFCache[6] = Code\_MNLR\_v3\_trimf((real\_T)Sum9, tmp\_5); tmp 5[0] = -3.0; $tmp_5[1] = 0.0;$ tmp 5[2] = 3.0;inputMFCache[7] = Code MNLR v3 trimf((real T)Sum9, tmp 5);  $tmp_5[0] = 0.0;$  $tmp_5[1] = 3.0;$  $tmp_5[2] = 6.0;$ inputMFCache[8] = Code\_MNLR\_v3\_trimf((real\_T)Sum9, tmp\_5); tmp 5[0] = 3.0; $tmp_5[1] = 6.0;$ tmp 5[2] = 9.0: inputMFCache[9] = Code MNLR v3 trimf((real T)Sum9, tmp 5); for (ruleID = 0; ruleID < 25; ruleID++) { Sum22 = inputMFCache[b 0[ruleID + 25] + 4] \* inputMFCache[b 0[ruleID] - 1];Sum 26 += Sum 22;rtb\_antecedentOutputs\_c[ruleID] = Sum22; ł /\* MATLAB Function: '<S18>/Evaluate Rule Consequents' \*/ Code EvaluateRuleConsequents p(Y, (real T)Sum9, rtb antecedentOutputs c, rtb aggregatedOutputs a, &Code MNLR v3 B.sf EvaluateRuleConsequents p); /\* MATLAB Function: '<S18>/Defuzzify Outputs' incorporates: \* MATLAB Function: '<S18>/Evaluate Rule Antecedents' \*/ if (Sum 26 == 0.0) { rtb defuzzifiedOutputs o idx 0 = 0.0; rtb\_defuzzifiedOutputs\_o\_idx\_1 = 0.0; rtb\_defuzzifiedOutputs\_o\_idx\_2 = 0.0; rtb\_defuzzifiedOutputs\_o\_idx\_3 = 0.0; } else { rtb defuzzifiedOutputs o idx 0 = 1.0 / Sum26 \* rtb aggregatedOutputs a[0]; rtb\_defuzzifiedOutputs\_o\_idx\_1 = 1.0 / Sum26 \* rtb\_aggregatedOutputs\_a[1]; rtb\_defuzzifiedOutputs\_o\_idx\_2 = 1.0 / Sum26 \* rtb\_aggregatedOutputs\_a[2]; rtb\_defuzzifiedOutputs\_o\_idx\_ $3 = 1.0 / Sum 26 * rtb_aggregatedOutputs_a[3];$ ł /\* End of MATLAB Function: '<S18>/Defuzzify Outputs' \*/ /\* End of Outputs for SubSystem: '<Root>/Fuzzy Logic Controller2' \*/ /\* Clock: '<S63>/Clock' incorporates: \* Clock: '<S54>/Clock' \* Clock: '<S55>/Clock' \* Clock: '<S64>/Clock' \* Clock: '<S65>/Clock' \* Clock: '<S66>/Clock' \* Clock: '<S67>/Clock' \* Clock: '<S68>/Clock' \* SimscapeExecutionBlock: '<S954>/OUTPUT 1 0' \* SimscapeExecutionBlock: '<S954>/STATE\_1' \*/

Y = Code\_MNLR\_v3\_M->Timing.t[0];

/\* Product: '<S63>/Product' incorporates:

```
* Clock: '<S63>/Clock'
* Constant: '<S63>/Constant'
* Step: '<S63>/Step'
* Sum: '<S63>/Sum'
*/
InputConversion_o_idx_0 = (real_T)!(Sum_fa_tmp < 19.5) * (Y - 19.5);
/* Product: '<S54>/Product' incorporates:
* Constant: '<S54>/Constant'
* Step: '<S54>/Step'
* Sum: '<S54>/Sum'
*/
Y f = (real T)!(Sum fa tmp < 14.5) * (Y - 14.5);
/* Product: '<S55>/Product' incorporates:
* Constant: '<S55>/Constant'
* Step: '<S55>/Step'
* Sum: '<S55>/Sum'
*/
rtb_defuzzifiedOutputs_g_idx_3 = (real_T)!(Sum_fa_tmp < 18.5) * (Y - 18.5);
/* Product: '<S64>/Product' incorporates:
* Constant: '<S64>/Constant'
* Step: '<S64>/Step'
* Sum: '<S64>/Sum'
*/
TmpSignalConversionAtSFunctio_k = (real_T)!(Sum_fa_tmp < 23.0) * (Y - 23.0);
/* Gain: '<Root>/Gain' incorporates:
* Constant: '<S65>/Constant'
* Product: '<S65>/Product'
* Step: '<S65>/Step'
* Sum: '<S65>/Sum'
*/
Sum26 = -((real T)!(Sum fa tmp < 8.0) * (Y - 8.0));
/* Product: '<S66>/Product' incorporates:
* Constant: '<S66>/Constant'
* Step: '<S66>/Step'
* Sum: '<S66>/Sum'
*/
InputConversion_o_idx_1 = (real_T)!(Sum_fa_tmp < 10.0) * (Y - 10.0);
/* Product: '<S67>/Product' incorporates:
* Constant: '<S67>/Constant'
* Step: '<S67>/Step'
* Sum: '<S67>/Sum'
*/
Sum35 = (real_T)!(Sum_fa_tmp < 13.0) * (Y - 13.0);
/* Product: '<S68>/Product' incorporates:
* Constant: '<S68>/Constant'
* Step: '<S68>/Step'
* Sum: '<S68>/Sum'
*/
InputConversion_o_idx_2 = (real_T)!(Sum_fa_tmp < 2.5) * (Y - 2.5);
/* Product: '<S69>/Product' incorporates:
* Clock: '<S69>/Clock'
* Constant: '<S69>/Constant'
* Step: '<S69>/Step'
* Sum: '<S69>/Sum'
*/
mVal = (real_T)!(Sum_fa_tmp < 8.0) * (Code_MNLR_v3_M->Timing.t[0] - 8.0);
/* Saturate: '<Root>/Saturation27' */
if (InputConversion_o_idx_0 > 0.45) {
```

```
InputConversion_o_idx_0 = 0.45;
} else if (InputConversion o idx 0 < 0.0) {
 InputConversion_o_idx_0 = 0.0;
/* Saturate: '<Root>/Saturation18' */
if (Y_f > 0.45) {
 Y f = 0.45;
else if (Y_f < 0.0) 
 Y_f = 0.0;
}
/* Saturate: '<Root>/Saturation19' */
if (rtb defuzzifiedOutputs g idx 3 > 0.45) {
 rtb defuzzifiedOutputs g idx 3 = 0.45;
} else if (rtb_defuzzifiedOutputs_g_idx_3 < 0.0) {
 rtb defuzzifiedOutputs g idx 3 = 0.0;
ł
/* Saturate: '<Root>/Saturation28' */
if (TmpSignalConversionAtSFunctio k > 0.45) {
 TmpSignalConversionAtSFunctio_k = 0.45;
\} else if (TmpSignalConversionAtSFunctio k < 0.0) {
 TmpSignalConversionAtSFunctio k = 0.0;
/* Outputs for Atomic SubSystem: '<Root>/Fuzzy Logic Controller25' */
/* DataTypeConversion: '<S19>/InputConversion' incorporates:
* Saturate: '<Root>/Saturation18'
* Saturate: '<Root>/Saturation19'
* Saturate: '<Root>/Saturation27'
* Saturate: '<Root>/Saturation28'
* Sum: '<Root>/Sum36'
* Sum: '<Root>/Sum37'
*/
InputConversion_o_idx_0 = ((Y_f - rtb_defuzzifiedOutputs_g_idx_3) +
 InputConversion_o_idx_0) - TmpSignalConversionAtSFunctio_k;
/* End of Outputs for SubSystem: '<Root>/Fuzzy Logic Controller25' */
/* Saturate: '<Root>/Saturation29' */
if (Sum 26 > 0.0) {
 Sum 26 = 0.0;
else if (Sum 26 < -0.3) 
 Sum 26 = -0.3;
}
/* Saturate: '<Root>/Saturation30' */
if (InputConversion o idx 1 > 0.45) {
InputConversion_o_idx_1 = 0.45;
} else if (InputConversion o idx 1 < 0.0) {
 InputConversion o idx 1 = 0.0;
/* Saturate: '<Root>/Saturation31' */
if (Sum35 > 0.15) {
 Sum35 = 0.15;
else if (Sum 35 < 0.0) 
 Sum35 = 0.0;
}
/* Outputs for Atomic SubSystem: '<Root>/Fuzzy Logic Controller25' */
/* DataTypeConversion: '<S19>/InputConversion' incorporates:
* Saturate: '<Root>/Saturation29'
* Saturate: '<Root>/Saturation30'
* Saturate: '<Root>/Saturation31'
* Sum: '<Root>/Sum40'
```

```
* Sum: '<Root>/Sum41'
*/
InputConversion_o_idx_1 = (Sum26 + InputConversion_o_idx_1) - Sum35;
/* End of Outputs for SubSystem: '<Root>/Fuzzy Logic Controller25' */
/* Saturate: '<Root>/Saturation32' */
if (InputConversion_o_idx_2 > 0.45) {
 InputConversion o idx 2 = 0.45;
} else if (InputConversion_o_idx_2 < 0.0) {
 InputConversion_o_idx_2 = 0.0;
}
/* Saturate: '<Root>/Saturation33' */
if (mVal > 0.45) {
 mVal = 0.45:
else if (mVal < 0.0) 
 mVal = 0.0;
ł
/* Outputs for Atomic SubSystem: '<Root>/Fuzzy Logic Controller25' */
/* DataTypeConversion: '<S19>/InputConversion' incorporates:
* Saturate: '<Root>/Saturation32'
* Saturate: '<Root>/Saturation33'
* Sum: '<Root>/Sum42'
*/
InputConversion o idx 2 -= mVal;
/* MATLAB Function: '<S19>/Evaluate Rule Antecedents' incorporates:
* DataTypeConversion: '<S19>/InputConversion'
*/
Sum 26 = 0.0;
tmp 5[0] = -1.0;
tmp_5[1] = -0.5;
tmp_5[2] = 0.0;
inputMFCache 0[0] = \text{Code MNLR v3 trimf(InputConversion o idx 0, tmp 5)};
tmp_5[0] = -0.5;
tmp_5[1] = 0.0;
tmp_5[2] = 0.5;
inputMFCache_0[1] = Code_MNLR_v3_trimf(InputConversion_o_idx_0, tmp_5);
tmp 5[0] = 0.0;
tmp_5[1] = 0.5;
tmp 5[2] = 1.0;
inputMFCache 0[2] = Code MNLR v_3 trimf(InputConversion o idx 0, tmp 5);
tmp_5[0] = -1.0;
tmp_5[1] = -0.5;
tmp 5[2] = 0.0;
inputMFCache_0[3] = Code_MNLR_v3_trimf(InputConversion_o_idx_1, tmp_5);
tmp 5[0] = -0.5;
tmp_5[1] = 0.0;
tmp_5[2] = 0.5;
inputMFCache 0[4] = Code MNLR v_3 trimf(InputConversion o idx 1, tmp 5);
tmp_5[0] = 0.0;
tmp_5[1] = 0.5;
tmp_5[2] = 1.0;
inputMFCache_0[5] = Code_MNLR_v3_trimf(InputConversion_o_idx_1, tmp_5);
tmp 5[0] = -1.0;
tmp_5[1] = -0.5;
tmp 5[2] = 0.0;
inputMFCache 0[6] = Code MNLR v_3 trimf(InputConversion o idx 2, tmp 5);
tmp_5[0] = -0.5;
tmp_5[1] = 0.0;
tmp_5[2] = 0.5;
```

```
inputMFCache_0[7] = Code_MNLR_v3_trimf(InputConversion_o_idx_2, tmp_5);
tmp 5[0] = 0.0;
tmp_5[1] = 0.5;
tmp_5[2] = 1.0;
inputMFCache_0[8] = Code_MNLR_v3_trimf(InputConversion_o_idx_2, tmp_5);
inputMFCache_0[9] = Code_MNLR_v3_trimf(0.0, i);
inputMFCache 0[10] = Code MNLR v3 trimf(0.0, k);
inputMFCache_0[11] = Code_MNLR_v3_trimf(0.0, g);
inputMFCache_0[12] = Code_MNLR_v3_trimf(0.0, j);
tmp 5[0] = 0.375;
tmp_5[1] = 0.5;
tmp 5[2] = 0.625;
inputMFCache 0[13] = \text{Code MNLR v3 trimf}(0.0, \text{tmp 5});
inputMFCache_0[14] = Code_MNLR_v3_trimf(0.0, e);
inputMFCache 0[15] = Code MNLR v3 trimf(0.0, d);
inputMFCache_0[16] = Code_MNLR_v3_trimf(0.0, i);
inputMFCache_0[17] = Code_MNLR_v3_trimf(0.0, h);
inputMFCache 0[18] = Code MNLR v3 trimf(0.0, g);
inputMFCache_0[19] = Code_MNLR_v3_trimf(0.0, f);
tmp 5[0] = 0.375;
tmp_5[1] = 0.5;
tmp_5[2] = 0.625;
inputMFCache 0[20] = Code MNLR v3 trimf(0.0, tmp 5);
inputMFCache_0[21] = Code_MNLR_v3_trimf(0.0, e);
inputMFCache_0[22] = Code_MNLR_v3_trimf(0.0, d);
/* MATLAB Function: '<S19>/Evaluate Rule Consequents' */
rtb_aggregatedOutputs_a[0] = 0.0;
rtb aggregatedOutputs a[1] = 0.0;
rtb aggregatedOutputs a[2] = 0.0;
rtb_aggregatedOutputs_a[3] = 0.0;
outputMFCache[0] = 0.375;
outputMFCache[1] = 0.1875;
outputMFCache[2] = 0.0;
outputMFCache[3] = -0.1875;
outputMFCache[4] = -0.375;
outputMFCache[5] = 0.375;
outputMFCache[6] = 0.1875;
outputMFCache[7] = 0.0;
outputMFCache[8] = -0.1875;
outputMFCache[9] = -0.375;
outputMFCache[10] = 0.375;
outputMFCache[11] = 0.1875;
outputMFCache[12] = 0.0;
outputMFCache[13] = -0.1875;
outputMFCache[14] = -0.375;
outputMFCache[15] = 0.375;
outputMFCache[16] = 0.1875;
outputMFCache[17] = 0.0;
outputMFCache[18] = -0.1875;
outputMFCache[19] = -0.375;
for (ruleID = 0; ruleID < 29; ruleID++) {
 /* MATLAB Function: '<S19>/Evaluate Rule Antecedents' */
 rtb antecedentOutputs b[ruleID] = 1.0;
 for (Sum9 = 0; Sum9 < 5; Sum9++) {
  InputConversion o idx 2 = rtb antecedentOutputs b[ruleID];
  b = b_2[29 * Sum9 + ruleID];
  mfIndex = (int32_T)fabs((real_T)b);
  if (mfIndex != 0) {
```

```
mVal = inputMFCache_0[(c_0[Sum9] + mfIndex) - 1];
   if (b < 0) {
    mVal = 1.0 - mVal;
   ł
   InputConversion o idx 2 *= mVal;
  }
  rtb antecedentOutputs b[ruleID] = InputConversion o idx 2;
 Sum26 += rtb_antecedentOutputs_b[ruleID];
 /* MATLAB Function: '<S19>/Evaluate Rule Consequents' */
 rtb_aggregatedOutputs_a[0] += outputMFCache[b_1[ruleID] - 1] *
  rtb antecedentOutputs b[ruleID];
 rtb_aggregatedOutputs_a[1] += outputMFCache[b 1[ruleID + 29] + 4] *
  rtb_antecedentOutputs_b[ruleID];
 rtb aggregatedOutputs a[2] += outputMFCache[b 1[ruleID + 58] + 9] *
  rtb_antecedentOutputs_b[ruleID];
 rtb_aggregatedOutputs_a[3] += outputMFCache[b_1[ruleID + 87] + 14] *
  rtb_antecedentOutputs_b[ruleID];
}
/* MATLAB Function: '<S19>/Defuzzify Outputs' incorporates:
* MATLAB Function: '<S19>/Evaluate Rule Antecedents'
*/
if (Sum 26 == 0.0) {
 rtb_defuzzifiedOutputs_g_idx_0 = 0.0;
 InputConversion_o_idx_2 = 0.0;
 InputConversion o idx 1 = 0.0;
rtb_defuzzifiedOutputs_g_idx_3 = 0.0;
} else {
 rtb defuzzifiedOutputs g idx 0 = 1.0 / \text{Sum}26 * \text{rtb} aggregatedOutputs a[0];
 InputConversion_o_idx_2 = 1.0 / \text{Sum26} * \text{rtb}_aggregatedOutputs}[1];
 InputConversion_o_idx_1 = 1.0 / Sum26 * rtb_aggregatedOutputs_a[2];
rtb_defuzzifiedOutputs_g_idx_3 = 1.0 / Sum26 * rtb_aggregatedOutputs_a[3];
/* End of MATLAB Function: '<S19>/Defuzzify Outputs' */
/* End of Outputs for SubSystem: '<Root>/Fuzzy Logic Controller25' */
/* TransportDelay: '<Root>/Transport Delay9' */
{
 real T **uBuffer = (real T**)
  &Code MNLR v3 DW.TransportDelay9 PWORK.TUbufferPtrs[0];
 real_T simTime = Code_MNLR_v3_M->Timing.t[0];
 real T tMinusDelay = simTime - 0.001;
 Code_MNLR_v3_B.TransportDelay9 = rt_TDelayInterpolate(
  tMinusDelay,
  0.0,
  *uBuffer.
  Code_MNLR_v3_DW.TransportDelay9_IWORK.CircularBufSize,
  &Code MNLR v3 DW.TransportDelay9 IWORK.Last,
  Code_MNLR_v3_DW.TransportDelay9_IWORK.Tail,
  Code_MNLR_v3_DW.TransportDelay9_IWORK.Head,
  0.0.
  0,
  0);
/* Sum: '<Root>/Sum18' incorporates:
* Constant: '<Root>/Constant3'
* Sum: '<Root>/Sum12'
*/
rtb_defuzzifiedOutputs_o_idx_0 = (0.25 - (rtb_defuzzifiedOutputs_o_idx_0 +
```

InputConversion\_o\_idx\_1)) - Code\_MNLR\_v3\_B.TransportDelay9; /\* Gain: '<S217>/Filter Coefficient' incorporates: \* Gain: '<S208>/Derivative Gain' \* Integrator: '<S209>/Filter' \* Sum: '<S209>/SumD' \*/ Code MNLR v3 B.FilterCoefficient = (1.56E-5 \* rtb defuzzifiedOutputs o idx 0- Code MNLR v3 X.Filter CSTATE) \* 100.0; /\* SimscapeInputBlock: '<S954>/INPUT\_10\_1\_1' incorporates: \* Gain: '<S219>/Proportional Gain' \* Integrator: '<S214>/Integrator' \* Sum: '<S223>/Sum' \*/ Code\_MNLR\_v3\_B.INPUT\_10\_1\_1[0] = (0.016 \* rtb\_defuzzifiedOutputs\_0\_idx\_0 + Code MNLR v3 X.Integrator CSTATE) + Code MNLR v3 B.FilterCoefficient; Code\_MNLR\_v3\_B.INPUT\_10\_1\_1[1] = 0.0; Code\_MNLR\_v3\_B.INPUT\_ $10_1_1[2] = 0.0;$ Code MNLR v3 B.INPUT 10 1 1[3] = 0.0;/\* TransportDelay: '<Root>/Transport Delay10' \*/ { real T \*\*uBuffer = (real T\*\*)&Code\_MNLR\_v3\_DW.TransportDelay10\_PWORK.TUbufferPtrs[0]; real T simTime = Code MNLR v3 M->Timing.t[0]; real T tMinusDelay = simTime - 0.001; Code\_MNLR\_v3\_B.TransportDelay10 = rt\_TDelayInterpolate( tMinusDelay, 0.0, \*uBuffer. Code MNLR v3 DW.TransportDelay10 IWORK.CircularBufSize, &Code\_MNLR\_v3\_DW.TransportDelay10\_IWORK.Last, Code MNLR v3 DW.TransportDelay10 IWORK.Tail, Code\_MNLR\_v3\_DW.TransportDelay10\_IWORK.Head, 0.0, 0, 0); /\* Sum: '<Root>/Sum28' incorporates: \* Constant: '<Root>/Constant3' \* Sum: '<Root>/Sum12' \*/  $rtb_defuzzifiedOutputs_o_idx_1 = (0.25 - (rtb_defuzzifiedOutputs_o_idx_1 +$ InputConversion\_o\_idx\_1)) - Code\_MNLR\_v3\_B.TransportDelay10; /\* Gain: '<S889>/Filter Coefficient' incorporates: \* Gain: '<S880>/Derivative Gain' \* Integrator: '<S881>/Filter' \* Sum: '<S881>/SumD' \*/ Code\_MNLR\_v3\_B.FilterCoefficient\_n = (1.56E-5 \*rtb\_defuzzifiedOutputs\_o\_idx\_1 - Code\_MNLR\_v3\_X.Filter\_CSTATE\_p) \* 100.0; /\* SimscapeInputBlock: '<S954>/INPUT\_11\_1\_1' incorporates: \* Gain: '<S891>/Proportional Gain' \* Integrator: '<S886>/Integrator' \* Sum: '<S895>/Sum' \*/ Code MNLR v3 B.INPUT 11 1 1[0] = (0.016 \* rtb defuzzifiedOutputs o idx 1 +Code\_MNLR\_v3\_X.Integrator\_CSTATE\_p) + Code\_MNLR\_v3\_B.FilterCoefficient\_n; Code\_MNLR\_v3\_B.INPUT\_11\_1\_1[1] = 0.0; Code\_MNLR\_v3\_B.INPUT\_11\_1\_1[2] = 0.0;

Code\_MNLR\_v3\_B.INPUT\_11\_1\_1[3] = 0.0; /\* Sum: '<Root>/Sum' incorporates: \* Sin: '<Root>/Sine Wave2' \* Step: '<Root>/Step14' \*/  $Sum_{fa} = sin(12.0 * Sum_{fa}tmp) * 110000.0 + (real_T)!(Sum_{fa}tmp < 1.0);$ /\* SimscapeInputBlock: '<S954>/INPUT 12 1 1' \*/ Code\_MNLR\_v3\_B.INPUT\_ $12_1_1[0] =$ Sum\_fa; Code\_MNLR\_v3\_B.INPUT\_12\_1\_1[1] = 0.0; Code\_MNLR\_v3\_B.INPUT\_12\_1\_1[2] = 0.0; Code\_MNLR\_v3\_B.INPUT\_ $12_1_1[3] = 0.0;$ /\* Sin: '<Root>/Sine Wave1' incorporates: \* Sin: '<Root>/Sine Wave3' \* Sin: '<Root>/Sine Wave4' \*/ Sum2 = sin(12.0 \* Code\_MNLR\_v3\_M->Timing.t[0]) \* 110000.0; /\* Sum: '<Root>/Sum20' incorporates: \* Sin: '<Root>/Sine Wave1' \* Step: '<Root>/Step5' \*/ mVal = (real T)!(Sum fa tmp < 1.0) + Sum2;/\* SimscapeInputBlock: '<S954>/INPUT\_13\_1\_1' incorporates: \* Gain: '<Root>/Gain3' \*/ Code\_MNLR\_v3\_B.INPUT\_13\_1\_1[0] = -mVal;Code\_MNLR\_v3\_B.INPUT\_13\_1\_1[1] = 0.0; Code\_MNLR\_v3\_B.INPUT\_13\_1\_1[2] = 0.0; Code MNLR v3 B.INPUT 13 1 1[3] = 0.0;/\* TransportDelay: '<Root>/Transport Delay11' \*/ { real T \*\*uBuffer = (real T\*\*)&Code\_MNLR\_v3\_DW.TransportDelay11\_PWORK.TUbufferPtrs[0]; real\_T simTime = Code\_MNLR\_v3\_M->Timing.t[0]; real T tMinusDelay = simTime - 0.001; Code\_MNLR\_v3\_B.TransportDelay11 = rt\_TDelayInterpolate( tMinusDelay, 0.0. \*uBuffer. Code MNLR v3 DW.TransportDelay11 IWORK.CircularBufSize, &Code\_MNLR\_v3\_DW.TransportDelay11\_IWORK.Last, Code MNLR v3 DW.TransportDelay11 IWORK.Tail, Code\_MNLR\_v3\_DW.TransportDelay11\_IWORK.Head, 0.0, 0, 0); /\* Sum: '<Root>/Sum10' incorporates: Constant: '<Root>/Constant3' \* Sum: '<Root>/Sum12' \*/ rtb\_defuzzifiedOutputs\_o\_idx\_2 = (0.25 - (rtb\_defuzzifiedOutputs\_o\_idx\_2 + InputConversion o idx 1)) - Code MNLR v3 B.TransportDelay11; /\* Gain: '<S265>/Filter Coefficient' incorporates: \* Gain: '<S256>/Derivative Gain' \* Integrator: '<S257>/Filter' \* Sum: '<S257>/SumD' \*/ Code\_MNLR\_v3\_B.FilterCoefficient\_l = (1.56E-5 \*

rtb\_defuzzifiedOutputs\_o\_idx\_2 - Code\_MNLR\_v3\_X.Filter\_CSTATE\_i) \* 100.0; /\* SimscapeInputBlock: '<S954>/INPUT 14 1 1' incorporates: \* Gain: '<S267>/Proportional Gain' \* Integrator: '<S262>/Integrator' \* Sum: '<S271>/Sum' \*/ Code MNLR v3 B.INPUT 14 1 1[0] = (0.016 \* rtb defuzzifiedOutputs o idx 2 +Code\_MNLR\_v3\_X.Integrator\_CSTATE\_n) + Code\_MNLR\_v3\_B.FilterCoefficient\_l; Code\_MNLR\_v3\_B.INPUT\_14\_1\_1[1] = 0.0; Code\_MNLR\_v3\_B.INPUT\_14\_1\_1[2] = 0.0; Code\_MNLR\_v3\_B.INPUT\_ $14_1_1[3] = 0.0;$ /\* SimscapeInputBlock: '<S954>/INPUT 15 1 1' incorporates: \* Gain: '<Root>/Gain3' \*/ Code\_MNLR\_v3\_B.INPUT\_ $15_1_1[0] = -mVal;$ Code\_MNLR\_v3\_B.INPUT\_ $15_1_1[1] = 0.0;$ Code\_MNLR\_v3\_B.INPUT\_ $15_1_1[2] = 0.0;$ Code MNLR v3 B.INPUT 15 1 1[3] = 0.0;/\* TransportDelay: '<Root>/Transport Delay12' \*/ { real T \*\*uBuffer = (real T\*\*)&Code\_MNLR\_v3\_DW.TransportDelay12\_PWORK.TUbufferPtrs[0]; real T simTime = Code MNLR v3 M->Timing.t[0]; real T tMinusDelay = simTime - 0.001; Code\_MNLR\_v3\_B.TransportDelay12 = rt\_TDelayInterpolate( tMinusDelay, 0.0, \*uBuffer. Code MNLR v3 DW.TransportDelay12 IWORK.CircularBufSize, &Code\_MNLR\_v3\_DW.TransportDelay12\_IWORK.Last, Code MNLR v3 DW.TransportDelay12 IWORK.Tail, Code\_MNLR\_v3\_DW.TransportDelay12\_IWORK.Head, 0.0, 0, 0); /\* Sum: '<Root>/Sum16' incorporates: \* Constant: '<Root>/Constant3' \* Sum: '<Root>/Sum12' \*/ rtb\_defuzzifiedOutputs\_o\_idx\_3 = (0.25 - (rtb\_defuzzifiedOutputs\_o\_idx\_3 + InputConversion\_o\_idx\_1)) - Code\_MNLR\_v3\_B.TransportDelay12; /\* Gain: '<S313>/Filter Coefficient' incorporates: \* Gain: '<S304>/Derivative Gain' \* Integrator: '<S305>/Filter' \* Sum: '<S305>/SumD' \*/ Code\_MNLR\_v3\_B.FilterCoefficient\_m = (1.56E-5 \*rtb\_defuzzifiedOutputs\_o\_idx\_3 - Code\_MNLR\_v3\_X.Filter\_CSTATE\_k) \* 100.0; /\* SimscapeInputBlock: '<S954>/INPUT\_16\_1\_1' incorporates: \* Gain: '<S315>/Proportional Gain' \* Integrator: '<S310>/Integrator' \* Sum: '<S319>/Sum' \*/ Code MNLR v3 B.INPUT 16 1 1[0] = (0.016 \* rtb defuzzifiedOutputs o idx 3 +Code\_MNLR\_v3\_X.Integrator\_CSTATE\_j) + Code\_MNLR\_v3\_B.FilterCoefficient\_m; Code\_MNLR\_v3\_B.INPUT\_16\_1\_1[1] = 0.0; Code\_MNLR\_v3\_B.INPUT\_ $16_1_1[2] = 0.0;$ 

```
Code_MNLR_v3_B.INPUT_16_1_1[3] = 0.0;
/* SimscapeInputBlock: '<S954>/INPUT 17 1 1' */
Code_MNLR_v3_B.INPUT_17_1_1[0] = mVal;
Code_MNLR_v3_B.INPUT_17_1_1[1] = 0.0;
Code_MNLR_v3_B.INPUT_17_1_1[2] = 0.0;
Code_MNLR_v3_B.INPUT_17_1_1[3] = 0.0;
/* Sum: '<Root>/Sum34' incorporates:
* Step: '<Root>/Step20'
*/
Sum34 = (real T)!(Sum fa tmp < 1.0) + Sum2;
/* SimscapeInputBlock: '<S954>/INPUT_18_1_1' */
Code MNLR v3 B.INPUT 18 1 1[0] = Sum34;
Code_MNLR_v3_B.INPUT_18_1_1[1] = 0.0;
Code_MNLR_v3_B.INPUT_18_1_1[2] = 0.0;
Code MNLR v3 B.INPUT 18 1 1[3] = 0.0;
/* Sum: '<Root>/Sum33' incorporates:
* Step: '<Root>/Step15'
* Step: '<Root>/Step7'
*/
if (Sum fa tmp < 0.5) {
ruleID = 0;
} else {
ruleID = 9;
if (Sum_fa_tmp < 1.0) {
Sum9 = 0;
} else {
Sum9 = -9;
ł
/* Outputs for Atomic SubSystem: '<Root>/Fuzzy Logic Controller3' */
/* DataTypeConversion: '<S20>/InputConversion' incorporates:
* Step: '<Root>/Step15'
* Step: '<Root>/Step7'
* Sum: '<Root>/Sum33'
*/
TmpSignalConversionAtSFunctio k = ruleID + Sum9;
/* MATLAB Function: '<S20>/Evaluate Rule Antecedents' incorporates:
* DataTypeConversion: '<S20>/InputConversion'
*/
Sum 26 = 0.0;
tmp_5[0] = -13.5;
tmp_5[1] = -9.0;
tmp_5[2] = -4.5;
inputMFCache[0] = Code MNLR v3 trimf(TmpSignalConversionAtSFunctio k, tmp 5);
tmp 5[0] = -9.0;
tmp_5[1] = -4.5;
tmp 5[2] = 0.0;
inputMFCache[1] = Code_MNLR_v3_trimf(TmpSignalConversionAtSFunctio_k, tmp_5);
inputMFCache[2] = Code_MNLR_v3_trimf(TmpSignalConversionAtSFunctio_k, c);
tmp_5[0] = 0.0;
tmp_5[1] = 4.5;
tmp 5[2] = 9.0;
inputMFCache[3] = Code_MNLR_v3_trimf(TmpSignalConversionAtSFunctio_k, tmp_5);
tmp 5[0] = 4.5;
tmp_5[1] = 9.0;
tmp_5[2] = 13.5;
inputMFCache[4] = Code_MNLR_v3_trimf(TmpSignalConversionAtSFunctio_k, tmp_5);
tmp_5[0] = -9.0;
```

 $tmp_5[1] = -6.0;$  $tmp_5[2] = -3.0;$ inputMFCache[5] = Code\_MNLR\_v3\_trimf(0.0, tmp\_5);  $tmp_5[0] = -6.0;$  $tmp_5[1] = -3.0;$  $tmp_5[2] = 0.0;$ inputMFCache[6] = Code MNLR v3 trimf(0.0, tmp 5);  $tmp_5[0] = -3.0;$  $tmp_5[1] = 0.0;$  $tmp_5[2] = 3.0;$ inputMFCache[7] = Code\_MNLR\_v3\_trimf(0.0, tmp\_5); tmp 5[0] = 0.0; $tmp_5[1] = 3.0;$  $tmp_5[2] = 6.0;$ inputMFCache[8] = Code MNLR v3 trimf(0.0, tmp 5);  $tmp_5[0] = 3.0;$  $tmp_5[1] = 6.0;$  $tmp_5[2] = 9.0;$ inputMFCache[9] = Code\_MNLR\_v3\_trimf(0.0, tmp\_5); for (ruleID = 0; ruleID < 25; ruleID++) {  $Sum22 = inputMFCache[b_0[ruleID + 25] + 4] * inputMFCache[b_0[ruleID] - 1];$ Sum26 += Sum22; rtb antecedentOutputs c[ruleID] = Sum22; /\* MATLAB Function: '<S20>/Evaluate Rule Consequents' \*/ Code MN EvaluateRuleConsequents(rtb antecedentOutputs c, rtb\_aggregatedOutputs\_a); /\* MATLAB Function: '<S20>/Defuzzify Outputs' incorporates: \* MATLAB Function: '<S20>/Evaluate Rule Antecedents' \*/ if (Sum 26 == 0.0) { TmpSignalConversionAtSFunctio\_k = 0.0; } else { TmpSignalConversionAtSFunctio k = 1.0 / Sum26 \* rtb aggregatedOutputs a[0]; /\* End of Outputs for SubSystem: '<Root>/Fuzzy Logic Controller3' \*/ /\* Sum: '<Root>/Sum30' \*/ rtb defuzzifiedOutputs ol = TmpSignalConversionAtSFunctio k +rtb defuzzifiedOutputs g idx 3; /\* Outputs for Atomic SubSystem: '<Root>/Fuzzy Logic Controller3' \*/ /\* MATLAB Function: '<S20>/Defuzzify Outputs' incorporates: \* MATLAB Function: '<S20>/Evaluate Rule Antecedents' \*/ if (Sum 26 == 0.0) { TmpSignalConversionAtSFunctio\_k = 0.0; } else { TmpSignalConversionAtSFunctio k = 1.0 / Sum26 \* rtb aggregatedOutputs a[1]; /\* End of Outputs for SubSystem: '<Root>/Fuzzy Logic Controller3' \*/ /\* Sum: '<Root>/Sum30' \*/ Sum6 = TmpSignalConversionAtSFunctio\_k + rtb\_defuzzifiedOutputs\_g\_idx\_3; /\* Outputs for Atomic SubSystem: '<Root>/Fuzzy Logic Controller3' \*/ /\* MATLAB Function: '<S20>/Defuzzify Outputs' incorporates: \* MATLAB Function: '<S20>/Evaluate Rule Antecedents' \*/ if (Sum 26 == 0.0) { TmpSignalConversionAtSFunctio\_k = 0.0; } else {

```
TmpSignalConversionAtSFunctio_k = 1.0 / Sum26 * rtb_aggregatedOutputs_a[2];
}
/* End of Outputs for SubSystem: '<Root>/Fuzzy Logic Controller3' */
/* Sum: '<Root>/Sum30' */
Sum 22 = TmpSignalConversionAtSFunctio k + rtb defuzzifiedOutputs g idx 3;
/* Outputs for Atomic SubSystem: '<Root>/Fuzzy Logic Controller3' */
/* MATLAB Function: '<S20>/Defuzzify Outputs' incorporates:
* MATLAB Function: '<S20>/Evaluate Rule Antecedents'
*/
if (Sum 26 == 0.0) {
 TmpSignalConversionAtSFunctio_k = 0.0;
} else {
 TmpSignalConversionAtSFunctio k = 1.0 / \text{Sum26} * \text{rtb} aggregatedOutputs a[3];
/* End of Outputs for SubSystem: '<Root>/Fuzzy Logic Controller3' */
/* TransportDelay: '<Root>/Transport Delay13' */
 real T **uBuffer = (real T**)
  &Code_MNLR_v3_DW.TransportDelay13_PWORK.TUbufferPtrs[0];
 real T simTime = Code MNLR v3 M->Timing.t[0];
 real T tMinusDelay = simTime - 0.001:
 Code_MNLR_v3_B.TransportDelay13 = rt_TDelayInterpolate(
  tMinusDelay,
  0.0,
  *uBuffer,
  Code MNLR v3 DW.TransportDelay13 IWORK.CircularBufSize,
  &Code_MNLR_v3_DW.TransportDelay13_IWORK.Last,
  Code MNLR v3 DW.TransportDelay13 IWORK.Tail,
  Code MNLR v3 DW.TransportDelay13 IWORK.Head,
  0.0.
  0,
  0);
/* Sum: '<Root>/Sum32' incorporates:
* Constant: '<Root>/Constant4'
*/
InputConversion o idx 1 = (0.25 - \text{rtb defuzzifiedOutputs ol}) -
 Code MNLR v3 B.TransportDelay13;
/* Gain: '<S361>/Filter Coefficient' incorporates:
* Gain: '<S352>/Derivative Gain'
* Integrator: '<S353>/Filter'
* Sum: '<S353>/SumD'
*/
Code MNLR v3 B.FilterCoefficient a = (1.56E-5 * InputConversion o idx 1 -
 Code MNLR v3 X.Filter CSTATE h) * 100.0;
/* SimscapeInputBlock: '<S954>/INPUT_19_1_1' incorporates:
* Gain: '<S363>/Proportional Gain'
* Integrator: '<S358>/Integrator'
* Sum: '<S367>/Sum'
*/
Code_MNLR_v3_B.INPUT_19_1_1[0] = (0.016 * InputConversion_o_idx_1 + 
 Code MNLR v3 X.Integrator CSTATE m) + Code MNLR v3 B.FilterCoefficient a;
Code_MNLR_v3_B.INPUT_19_1_1[1] = 0.0;
Code_MNLR_v3_B.INPUT_19_1_1[2] = 0.0;
Code MNLR v3 B.INPUT 19 1 1[3] = 0.0;
/* SimscapeInputBlock: '<S954>/INPUT_1_1_1' */
Code_MNLR_v3_B.INPUT_1_1_1[0] = Sum_fa;
Code_MNLR_v3_B.INPUT_1_1_1[1] = 0.0;
```

Code\_MNLR\_v3\_B.INPUT\_1\_1\_1[2] = 0.0; Code MNLR v3 B.INPUT 1 1 1[3] = 0.0;/\* TransportDelay: '<Root>/Transport Delay14' \*/ real T \*\*uBuffer = (real T\*\*)&Code\_MNLR\_v3\_DW.TransportDelay14\_PWORK.TUbufferPtrs[0]; real T simTime = Code MNLR v3 M->Timing.t[0]; real\_T tMinusDelay = simTime - 0.001; Code\_MNLR\_v3\_B.TransportDelay14 = rt\_TDelayInterpolate( tMinusDelay, 0.0, \*uBuffer. Code MNLR v3 DW.TransportDelay14 IWORK.CircularBufSize, &Code\_MNLR\_v3\_DW.TransportDelay14\_IWORK.Last, Code MNLR v3 DW.TransportDelay14 IWORK.Tail, Code\_MNLR\_v3\_DW.TransportDelay14\_IWORK.Head, 0.0, 0, 0); /\* Sum: '<Root>/Sum35' incorporates: \* Constant: '<Root>/Constant4' \*/ Sum35 = (0.25 - Sum6) - Code\_MNLR\_v3\_B.TransportDelay14; /\* Gain: '<S505>/Filter Coefficient' incorporates: \* Gain: '<S496>/Derivative Gain' \* Integrator: '<S497>/Filter' \* Sum: '<S497>/SumD' \*/ Code\_MNLR\_v3\_B.FilterCoefficient\_p = (1.56E-5 \* Sum35 -Code MNLR v3 X.Filter CSTATE a) \* 100.0; /\* SimscapeInputBlock: '<S954>/INPUT\_20\_1\_1' incorporates: \* Gain: '<S507>/Proportional Gain' \* Integrator: '<S502>/Integrator' \* Sum: '<S511>/Sum' \*/  $Code_MNLR_v3_B.INPUT_20_1_1[0] = (0.016 * Sum 35 +$ Code MNLR v3 X.Integrator CSTATE 1) + Code MNLR v3 B.FilterCoefficient p; Code MNLR v3 B.INPUT 20 1 1[1] = 0.0;Code\_MNLR\_v3\_B.INPUT\_20\_1\_1[2] = 0.0; Code\_MNLR\_v3\_B.INPUT\_20\_1\_1[3] = 0.0; /\* SimscapeInputBlock: '<S954>/INPUT\_21\_1\_1' incorporates: \* Gain: '<Root>/Gain4' \*/ Code \_MNLR\_v3\_B.INPUT\_21\_1\_1[0] = -Sum34; Code\_MNLR\_v3\_B.INPUT\_21\_1[1] = 0.0; Code\_MNLR\_v3\_B.INPUT\_21\_1\_1[2] = 0.0; Code\_MNLR\_v3\_B.INPUT\_21\_1\_1[3] = 0.0; /\* TransportDelay: '<Root>/Transport Delay15' \*/ { real\_T \*\*uBuffer = (real\_T\*\*) &Code MNLR v3 DW.TransportDelay15 PWORK.TUbufferPtrs[0]; real T simTime = Code MNLR v3 M->Timing.t[0]; real T tMinusDelay = simTime - 0.001; Code MNLR v3 B.TransportDelay15 = rt TDelayInterpolate( tMinusDelay, 0.0, \*uBuffer,

Code\_MNLR\_v3\_DW.TransportDelay15\_IWORK.CircularBufSize, &Code MNLR v3 DW.TransportDelay15 IWORK.Last, Code\_MNLR\_v3\_DW.TransportDelay15\_IWORK.Tail, Code\_MNLR\_v3\_DW.TransportDelay15\_IWORK.Head, 0.0. 0, 0); } /\* Sum: '<Root>/Sum29' incorporates: \* Constant: '<Root>/Constant4' \*/ InputConversion o idx 0 = (0.25 - Sum22) - Code MNLR v3 B.TransportDelay15;/\* Gain: '<S409>/Filter Coefficient' incorporates: \* Gain: '<S400>/Derivative Gain' \* Integrator: '<S401>/Filter' \* Sum: '<S401>/SumD' \*/ Code\_MNLR\_v3\_B.FilterCoefficient\_i =  $(1.56E-5 * InputConversion_o_idx_0 -$ Code\_MNLR\_v3\_X.Filter\_CSTATE\_g) \* 100.0; /\* SimscapeInputBlock: '<S954>/INPUT 22 1 1' incorporates: \* Gain: '<S411>/Proportional Gain' \* Integrator: '<S406>/Integrator' \* Sum: '<S415>/Sum' \*/  $Code_MNLR_v3_B.INPUT_22_1_1[0] = (0.016 * InputConversion_o_idx_0 +$ Code\_MNLR\_v3\_X.Integrator\_CSTATE\_ma) + Code\_MNLR\_v3\_B.FilterCoefficient\_i; Code\_MNLR\_v3\_B.INPUT\_22\_1\_1[1] = 0.0; Code MNLR v3 B.INPUT 22 1 1[2] = 0.0;Code\_MNLR\_v3\_B.INPUT\_22\_1\_1[3] = 0.0; /\* Step: '<Root>/Step31' \*/ if (Sum fa tmp < 0.5) { ruleID = 0; } else { ruleID = 9; } /\* Step: '<Root>/Step30' \*/ if (Sum fa tmp < 1.0) { Sum9 = 0;} else { Sum9 = -9;} /\* Step: '<Root>/Step33' \*/ if (Sum\_fa\_tmp < 2.5) { Sum 26 = 0.0;} else { Sum 26 = -4.5;ł /\* Step: '<Root>/Step32' \*/ if (Sum\_fa\_tmp < 3.0) { Sum22 = 0.0;} else { Sum22 = 4.5;ł /\* Step: '<Root>/Step3' \*/ if (Sum fa tmp < 3.5) { mfIndex = 0; } else { mfIndex = -6;

```
}
/* Step: '<Root>/Step2' */
if (Sum_fa_tmp < 4.0) {
 tmp_6 = 0;
} else {
tmp_6 = 6;
}
/* Sum: '<Root>/Sum14' incorporates:
* Step: '<Root>/Step2'
* Step: '<Root>/Step3'
* Step: '<Root>/Step30'
* Step: '<Root>/Step31'
* Step: '<Root>/Step32'
* Step: '<Root>/Step33'
* Sum: '<Root>/Sum11'
* Sum: '<Root>/Sum54'
* Sum: '<Root>/Sum55'
*/
Y_f = ((real_T)(ruleID + Sum9) + (Sum26 + Sum22)) + (real_T)(mfIndex + tmp_6);
/* Step: '<Root>/Step9' */
if (Sum_fa_tmp < 1.5) {
ruleID = 0;
} else {
ruleID = -6;
ł
/* Step: '<Root>/Step8' */
if (Sum_fa_tmp < 2.0) {
 Sum9 = 0;
} else {
 Sum9 = 6;
}
/* Step: '<Root>/Step35' */
if (Sum_fa_tmp < 2.5) {
 mfIndex = 0;
} else {
 mfIndex = 3;
ł
/* Step: '<Root>/Step34' */
if (Sum_fa_tmp < 3.0) {
 tmp_6 = 0;
} else {
tmp_6 = -3;
}
/* Step: '<Root>/Step11' */
if (Sum_fa_tmp < 3.5) {
 tmp_7 = 0;
} else {
tmp_7 = -5;
}
/* Step: '<Root>/Step10' */
if (Sum_fa_tmp < 4.0) {
 tmp 8 = 0;
} else {
tmp_8 = 5;
ł
/* Sum: '<Root>/Sum52' incorporates:
* Step: '<Root>/Step10'
* Step: '<Root>/Step11'
```

\* Step: '<Root>/Step34' \* Step: '<Root>/Step35' \* Step: '<Root>/Step8' \* Step: '<Root>/Step9' \* Sum: '<Root>/Sum13' \* Sum: '<Root>/Sum15' \* Sum: '<Root>/Sum53' \*/  $Sum9 = ((((ruleID + Sum9) + mfIndex) + tmp_6) + tmp_7) + tmp_8;$ /\* Outputs for Atomic SubSystem: '<Root>/Fuzzy Logic Controller4' \*/ /\* MATLAB Function: '<S21>/Evaluate Rule Antecedents' incorporates: \* SignalConversion generated from: '<S133>/ SFunction ' \*/ Sum 26 = 0.0; $tmp_5[0] = -13.5;$  $tmp_5[1] = -9.0;$  $tmp_5[2] = -4.5;$ inputMFCache[0] = Code\_MNLR\_v3\_trimf(Y\_f, tmp\_5);  $tmp_5[0] = -9.0;$ tmp 5[1] = -4.5;tmp 5[2] = 0.0;inputMFCache[1] = Code\_MNLR\_v3\_trimf(Y\_f, tmp\_5); inputMFCache[2] = Code MNLR v3 trimf(Y f, c);  $tmp_5[0] = 0.0;$  $tmp_5[1] = 4.5;$  $tmp_5[2] = 9.0;$ inputMFCache[3] = Code\_MNLR\_v3\_trimf(Y\_f, tmp\_5); tmp 5[0] = 4.5; $tmp_5[1] = 9.0;$  $tmp_5[2] = 13.5;$ inputMFCache[4] = Code MNLR v3 trimf(Y f, tmp 5);  $tmp_5[0] = -9.0;$  $tmp_5[1] = -6.0;$  $tmp_5[2] = -3.0;$ inputMFCache[5] = Code\_MNLR\_v3\_trimf((real\_T)Sum9, tmp\_5); tmp 5[0] = -6.0; $tmp_5[1] = -3.0;$  $tmp_5[2] = 0.0;$ inputMFCache[6] = Code MNLR v3 trimf((real T)Sum9, tmp 5);  $tmp_5[0] = -3.0;$  $tmp_5[1] = 0.0;$  $tmp_5[2] = 3.0;$ inputMFCache[7] = Code\_MNLR\_v3\_trimf((real\_T)Sum9, tmp\_5); tmp 5[0] = 0.0; $tmp_5[1] = 3.0;$  $tmp_5[2] = 6.0;$ inputMFCache[8] = Code MNLR v3 trimf((real T)Sum9, tmp 5);  $tmp_5[0] = 3.0;$  $tmp_5[1] = 6.0;$  $tmp_5[2] = 9.0;$ inputMFCache[9] = Code\_MNLR\_v3\_trimf((real\_T)Sum9, tmp\_5); for (ruleID = 0; ruleID < 25; ruleID++) {  $Sum22 = inputMFCache[b_0[ruleID + 25] + 4] * inputMFCache[b_0[ruleID] - 1];$ Sum26 += Sum22; rtb antecedentOutputs c[ruleID] = Sum22; /\* MATLAB Function: '<S21>/Evaluate Rule Consequents' \*/ Code\_\_EvaluateRuleConsequents\_p(Y\_f, (real\_T)Sum9, rtb\_antecedentOutputs\_c,

179

rtb\_aggregatedOutputs\_a, &Code\_MNLR\_v3\_B.sf\_EvaluateRuleConsequents\_l); /\* MATLAB Function: '<S21>/Defuzzify Outputs' incorporates: \* MATLAB Function: '<S21>/Evaluate Rule Antecedents' \*/ if (Sum 26 == 0.0) { Sum22 = 0.0;} else { Sum22 = 1.0 / Sum26 \* rtb\_aggregatedOutputs\_a[0]; } /\* End of Outputs for SubSystem: '<Root>/Fuzzy Logic Controller4' \*/ /\* Sum: '<Root>/Sum24' \*/ Y f = Sum22 + rtb defuzzifiedOutputs g idx 0; /\* Outputs for Atomic SubSystem: '<Root>/Fuzzy Logic Controller4' \*/ /\* MATLAB Function: '<S21>/Defuzzify Outputs' incorporates: \* MATLAB Function: '<S21>/Evaluate Rule Antecedents' \*/ if (Sum 26 == 0.0) { Sum22 = 0.0;} else { Sum22 = 1.0 / Sum26 \* rtb aggregatedOutputs a[1]; ł /\* End of Outputs for SubSystem: '<Root>/Fuzzy Logic Controller4' \*/ /\* Sum: '<Root>/Sum24' \*/ Sum5 = Sum22 + rtb\_defuzzifiedOutputs\_g\_idx\_0; /\* Outputs for Atomic SubSystem: '<Root>/Fuzzy Logic Controller4' \*/ /\* MATLAB Function: '<S21>/Defuzzify Outputs' incorporates: \* MATLAB Function: '<S21>/Evaluate Rule Antecedents' \*/ if (Sum 26 == 0.0) { Sum22 = 0.0;} else { Sum22 = 1.0 / Sum26 \* rtb\_aggregatedOutputs\_a[2]; /\* End of Outputs for SubSystem: '<Root>/Fuzzy Logic Controller4' \*/ /\* Sum: '<Root>/Sum24' \*/ Sum6 = Sum22 + rtb defuzzifiedOutputs g idx 0; /\* Outputs for Atomic SubSystem: '<Root>/Fuzzy Logic Controller4' \*/ /\* MATLAB Function: '<S21>/Defuzzify Outputs' incorporates: \* MATLAB Function: '<S21>/Evaluate Rule Antecedents' \*/ if (Sum 26 == 0.0) { Sum22 = 0.0;} else { Sum22 = 1.0 / Sum26 \* rtb aggregatedOutputs a[3]; /\* End of Outputs for SubSystem: '<Root>/Fuzzy Logic Controller4' \*/ /\* TransportDelay: '<Root>/Transport Delay1' \*/ ł real T \*\*uBuffer = (real T\*\*)&Code\_MNLR\_v3\_DW.TransportDelay1\_PWORK.TUbufferPtrs[0]; real\_T simTime = Code\_MNLR\_v3\_M->Timing.t[0]; real\_T tMinusDelay = simTime - 0.001; Code MNLR v3 B.TransportDelav1 = rt TDelavInterpolate( tMinusDelay, 0.0, \*uBuffer. Code\_MNLR\_v3\_DW.TransportDelay1\_IWORK.CircularBufSize, &Code\_MNLR\_v3\_DW.TransportDelay1\_IWORK.Last,
Code\_MNLR\_v3\_DW.TransportDelay1\_IWORK.Tail, Code\_MNLR\_v3\_DW.TransportDelay1\_IWORK.Head, 0.0, 0. 0);ł /\* Sum: '<Root>/Sum4' incorporates: Constant: '<Root>/Constant2' \*/ Y f = (0.25 - Y f) - Code MNLR v3 B.TransportDelay1; /\* Gain: '<S553>/Filter Coefficient' incorporates: \* Gain: '<S544>/Derivative Gain' \* Integrator: '<S545>/Filter' \* Sum: '<S545>/SumD' \*/ Code\_MNLR\_v3\_B.FilterCoefficient\_ab =  $(1.56E-5 * Y_f -$ Code\_MNLR\_v3\_X.Filter\_CSTATE\_gy) \* 100.0; /\* SimscapeInputBlock: '<S954>/INPUT 23 1 1' incorporates: \* Gain: '<S555>/Proportional Gain' \* Integrator: '<S550>/Integrator' \* Sum: '<S559>/Sum' \*/ Code MNLR v3 B.INPUT 23 1 1[0] = (0.016 \* Y f +Code\_MNLR\_v3\_X.Integrator\_CSTATE\_js) + Code\_MNLR\_v3\_B.FilterCoefficient\_ab; Code\_MNLR\_v3\_B.INPUT\_23\_1\_1[1] = 0.0; Code\_MNLR\_v3\_B.INPUT\_23\_1\_1[2] = 0.0; Code\_MNLR\_v3\_B.INPUT\_23\_1\_1[3] = 0.0; /\* SimscapeInputBlock: '<S954>/INPUT 24 1 1' incorporates: \* Gain: '<Root>/Gain4' \*/ Code\_MNLR\_v3\_B.INPUT\_24\_1\_1[0] = -Sum34; Code\_MNLR\_v3\_B.INPUT\_24\_1\_1[1] = 0.0; Code\_MNLR\_v3\_B.INPUT\_24\_1\_1[2] = 0.0; Code MNLR v3 B.INPUT 24 1 1[3] = 0.0;/\* TransportDelay: '<Root>/Transport Delay16' \*/ { real\_T \*\*uBuffer = (real T\*\*) &Code MNLR v3 DW.TransportDelay16 PWORK.TUbufferPtrs[0]; real T simTime = Code MNLR v3 M->Timing.t[0]; real\_T tMinusDelay = simTime - 0.001; Code\_MNLR\_v3\_B.TransportDelay16 = rt\_TDelayInterpolate( tMinusDelay, 0.0, \*uBuffer, Code MNLR v3 DW.TransportDelay16 IWORK.CircularBufSize, &Code\_MNLR\_v3\_DW.TransportDelay16\_IWORK.Last, Code MNLR v3 DW.TransportDelay16 IWORK.Tail, Code\_MNLR\_v3\_DW.TransportDelay16\_IWORK.Head, 0.0, 0. 0); /\* Sum: '<Root>/Sum31' incorporates: \* Constant: '<Root>/Constant4' \* Sum: '<Root>/Sum30' \*/  $rtb_defuzzifiedOutputs_g_idx_3 = (0.25 - (TmpSignalConversionAtSFunctio_k +$ rtb\_defuzzifiedOutputs\_g\_idx\_3)) - Code\_MNLR\_v3\_B.TransportDelay16;

- \* Gain: '<S448>/Derivative Gain'
- \* Integrator: '<S449>/Filter'
- \* Sum: '<S449>/SumD'

\*/

```
Code_MNLR_v3_B.FilterCoefficient_e = (1.56E-5 *
 rtb defuzzifiedOutputs g idx 3 - Code MNLR v3 X.Filter CSTATE ay) * 100.0;
/* SimscapeInputBlock: '<S954>/INPUT_25_1_1' incorporates:
* Gain: '<S459>/Proportional Gain'
* Integrator: '<S454>/Integrator'
* Sum: '<S463>/Sum'
*/
Code MNLR v3 B.INPUT 25 1 1[0] = (0.016 * \text{rtb} \text{ defuzzifiedOutputs g idx } 3 +
 Code_MNLR_v3_X.Integrator_CSTATE_a) + Code_MNLR_v3_B.FilterCoefficient_e;
Code_MNLR_v3_B.INPUT_25_1_1[1] = 0.0;
Code_MNLR_v3_B.INPUT_25_1_1[2] = 0.0;
Code_MNLR_v3_B.INPUT_25_1_1[3] = 0.0;
/* SimscapeInputBlock: '<S954>/INPUT 26 1 1' */
Code_MNLR_v3_B.INPUT_26_1_1[0] = Sum34;
Code MNLR v3 B.INPUT 26 1 1[1] = 0.0;
Code_MNLR_v3_B.INPUT_26_1_1[2] = 0.0;
Code_MNLR_v3_B.INPUT_26_1_1[3] = 0.0;
/* TransportDelay: '<Root>/Transport Delay2' */
{
 real_T **uBuffer = (real_T**)
  &Code MNLR v3 DW.TransportDelay2 PWORK.TUbufferPtrs[0];
 real_T simTime = Code_MNLR_v3_M->Timing.t[0];
 real T tMinusDelay = simTime - 0.001;
 Code_MNLR_v3_B.TransportDelay2 = rt_TDelayInterpolate(
  tMinusDelay,
  0.0.
  *uBuffer.
  Code_MNLR_v3_DW.TransportDelay2_IWORK.CircularBufSize,
  &Code MNLR v3 DW.TransportDelay2 IWORK.Last,
  Code_MNLR_v3_DW.TransportDelay2_IWORK.Tail,
  Code MNLR v3 DW.TransportDelay2 IWORK.Head,
  0.0,
  0,
  0);
}
/* Sum: '<Root>/Sum1' incorporates:
* Constant: '<Root>/Constant2'
*/
Sum34 = (0.25 - Sum5) - Code MNLR v3 B.TransportDelay2;
/* Gain: '<S169>/Filter Coefficient' incorporates:
* Gain: '<S160>/Derivative Gain'
* Integrator: '<S161>/Filter'
* Sum: '<S161>/SumD'
*/
Code_MNLR_v3_B.FilterCoefficient_ni = (1.56E-5 * Sum34 -
 Code_MNLR_v3_X.Filter_CSTATE_n) * 100.0;
/* SimscapeInputBlock: '<S954>/INPUT 27 1 1' incorporates:
* Gain: '<S171>/Proportional Gain'
* Integrator: '<S166>/Integrator'
* Sum: '<S175>/Sum'
*/
Code_MNLR_v3_B.INPUT_27_1_1[0] = (0.016 * Sum34 +
 Code_MNLR_v3_X.Integrator_CSTATE_f) + Code_MNLR_v3_B.FilterCoefficient_ni;
```

Code\_MNLR\_v3\_B.INPUT\_27\_1\_1[1] = 0.0; Code MNLR v3 B.INPUT 27 1 1[2] = 0.0;Code\_MNLR\_v3\_B.INPUT\_27\_1\_1[3] = 0.0; /\* Sum: '<Root>/Sum5' incorporates: \* Step: '<Root>/Step19' \*/ Sum5 = (real T)!(Sum fa tmp < 1.0) + Sum2;/\* SimscapeInputBlock: '<S954>/INPUT\_28\_1\_1' \*/ Code\_MNLR\_v3\_B.INPUT\_28\_1\_1[0] = Sum5; Code\_MNLR\_v3\_B.INPUT\_28\_1\_1[1] = 0.0; Code\_MNLR\_v3\_B.INPUT\_ $28_1_1[2] = 0.0;$ Code MNLR v3 B.INPUT 28 1 1[3] = 0.0;/\* SimscapeInputBlock: '<S954>/INPUT 29 1 1' incorporates: \* Gain: '<Root>/Gain2' \*/ Code\_MNLR\_v3\_B.INPUT\_29\_1\_1[0] = -Sum\_fa; Code\_MNLR\_v3\_B.INPUT\_29\_1\_1[1] = 0.0; Code MNLR v3 B.INPUT 29 1 1[2] = 0.0;Code\_MNLR\_v3\_B.INPUT\_29\_1\_1[3] = 0.0; /\* SimscapeInputBlock: '<S954>/INPUT 2 1 1' \*/ Code\_MNLR\_v3\_B.INPUT\_2\_1\_1[0] = Sum5; Code\_MNLR\_v3\_B.INPUT\_2\_1\_1[1] = 0.0; Code MNLR v3 B.INPUT 2 1 1[2] = 0.0;Code\_MNLR\_v3\_B.INPUT\_2\_1\_1[3] = 0.0; /\* TransportDelay: '<Root>/Transport Delay3' \*/ { real\_T \*\*uBuffer = (real\_T\*\*) &Code MNLR v3 DW.TransportDelay3 PWORK.TUbufferPtrs[0]; real T simTime = Code MNLR v3 M->Timing.t[0]; real\_T tMinusDelay = simTime - 0.001; Code MNLR v3 B.TransportDelay3 = rt TDelayInterpolate( tMinusDelay, 0.0, \*uBuffer, Code\_MNLR\_v3\_DW.TransportDelay3\_IWORK.CircularBufSize, &Code MNLR v3 DW.TransportDelay3 IWORK.Last, Code MNLR v3 DW.TransportDelay3 IWORK.Tail, Code MNLR v3 DW.TransportDelay3 IWORK.Head, 0.0, 0. 0);} /\* Sum: '<Root>/Sum2' incorporates: Constant: '<Root>/Constant2' \*/ Sum2 = (0.25 - Sum6) - Code MNLR v3 B.TransportDelav3;/\* Gain: '<S601>/Filter Coefficient' incorporates: \* Gain: '<S592>/Derivative Gain' \* Integrator: '<S593>/Filter' \* Sum: '<S593>/SumD' \*/ Code\_MNLR\_v3\_B.FilterCoefficient\_k = (1.56E-5 \* Sum2 -Code MNLR v3 X.Filter CSTATE o) \* 100.0; /\* SimscapeInputBlock: '<S954>/INPUT\_30\_1\_1' incorporates: \* Gain: '<S603>/Proportional Gain' \* Integrator: '<S598>/Integrator' \* Sum: '<S607>/Sum' \*/ Code\_MNLR\_v3\_B.INPUT\_30\_1\_1[0] = (0.016 \* Sum2 +

Code\_MNLR\_v3\_X.Integrator\_CSTATE\_i) + Code\_MNLR\_v3\_B.FilterCoefficient\_k; Code MNLR v3 B.INPUT 30 1 1[1] = 0.0;Code\_MNLR\_v3\_B.INPUT\_30\_1\_1[2] = 0.0; Code\_MNLR\_v3\_B.INPUT\_30\_1\_1[3] = 0.0; /\* SimscapeInputBlock: '<S954>/INPUT 31 1 1' incorporates: \* Gain: '<Root>/Gain2' \*/ Code\_MNLR\_v3\_B.INPUT\_31\_1\_1[0] = -Sum\_fa; Code\_MNLR\_v3\_B.INPUT\_31\_1\_1[1] = 0.0; Code\_MNLR\_v3\_B.INPUT\_31\_1\_1[2] = 0.0; Code\_MNLR\_v3\_B.INPUT\_31\_1\_1[3] = 0.0; /\* TransportDelay: '<Root>/Transport Delay4' \*/ { real\_T \*\*uBuffer = (real\_T\*\*) &Code MNLR v3 DW.TransportDelay4 PWORK.TUbufferPtrs[0]; real\_T simTime = Code\_MNLR\_v3\_M->Timing.t[0]; real T tMinusDelay = simTime - 0.001; Code\_MNLR\_v3\_B.TransportDelay4 = rt\_TDelayInterpolate( tMinusDelay, 0.0. \*uBuffer. Code\_MNLR\_v3\_DW.TransportDelay4\_IWORK.CircularBufSize, &Code MNLR v3 DW.TransportDelay4 IWORK.Last, Code\_MNLR\_v3\_DW.TransportDelay4\_IWORK.Tail, Code\_MNLR\_v3\_DW.TransportDelay4\_IWORK.Head, 0.0, 0, 0); } /\* Sum: '<Root>/Sum3' incorporates: \* Constant: '<Root>/Constant2' \* Sum: '<Root>/Sum24' \*/ rtb defuzzifiedOutputs g idx 0 = (0.25 - (Sum 22 +rtb\_defuzzifiedOutputs\_g\_idx\_0)) - Code\_MNLR\_v3\_B.TransportDelay4; /\* Gain: '<S649>/Filter Coefficient' incorporates: \* Gain: '<S640>/Derivative Gain' \* Integrator: '<S641>/Filter' \* Sum: '<S641>/SumD' \*/ Code MNLR v3 B.FilterCoefficient me = (1.56E-5 \*rtb\_defuzzifiedOutputs\_g\_idx\_0 - Code\_MNLR\_v3\_X.Filter\_CSTATE\_pb) \* 100.0; /\* SimscapeInputBlock: '<S954>/INPUT\_32\_1\_1' incorporates: \* Gain: '<S651>/Proportional Gain' \* Integrator: '<S646>/Integrator' \* Sum: '<S655>/Sum' \*/ Code\_MNLR\_v3\_B.INPUT\_32\_1\_1[0] = (0.016 \* rtb\_defuzzifiedOutputs\_g\_idx\_0 + Code\_MNLR\_v3\_X.Integrator\_CSTATE\_k) + Code\_MNLR\_v3\_B.FilterCoefficient\_me; Code\_MNLR\_v3\_B.INPUT\_32\_1\_1[1] = 0.0; Code\_MNLR\_v3\_B.INPUT\_32\_1\_1[2] = 0.0; Code MNLR v3 B.INPUT 32 1 1[3] = 0.0;/\* TransferFcn: '<S103>/Transfer Fcn7' \*/ Sum\_fa = 1.5384615384615383 \* Code\_MNLR\_v3\_X.TransferFcn7\_CSTATE; /\* SimscapeInputBlock: '<S954>/INPUT 33 1 1' \*/ Code\_MNLR\_v3\_B.INPUT\_33\_1\_1[0] = Sum\_fa; Code\_MNLR\_v3\_B.INPUT\_33\_1\_1[1] = 0.0; Code\_MNLR\_v3\_B.INPUT\_33\_1\_1[2] = 0.0;

```
Code_MNLR_v3_B.INPUT_33_1_1[3] = 0.0;
/* Sum: '<Root>/Sum17' incorporates:
* Step: '<Root>/Step17'
* Step: '<Root>/Step18'
* Step: '<Root>/Step22'
* Step: '<Root>/Step23'
* Step: '<Root>/Step24'
* Step: '<Root>/Step25'
*/
if (Sum_fa_tmp < 0.5) {
ruleID = 0;
} else {
ruleID = 9;
if (Sum_fa_tmp < 1.0) {
 Sum9 = 0;
} else {
 Sum9 = -9;
if (Sum fa tmp < 2.5) {
 Sum 26 = 0.0;
} else {
 Sum 26 = -4.5;
if (Sum_fa_tmp < 3.0) {
 Sum22 = 0.0;
} else {
 Sum22 = 4.5;
ļ
if (Sum_fa_tmp < 3.5) {
 mfIndex = 0;
} else {
 mfIndex = -6;
if (Sum_fa_tmp < 4.0) {
 tmp 6 = 0;
} else {
 tmp_6 = 6;
}
/* Outputs for Atomic SubSystem: '<Root>/Fuzzy Logic Controller1' */
/* DataTypeConversion: '<S17>/InputConversion' incorporates:
* Step: '<Root>/Step17'
* Step: '<Root>/Step18'
* Step: '<Root>/Step22'
* Step: '<Root>/Step23'
* Step: '<Root>/Step24'
* Step: '<Root>/Step25'
  Sum: '<Root>/Sum17'
*
* Sum: '<Root>/Sum21'
* Sum: '<Root>/Sum27'
* Sum: '<Root>/Sum7'
*/
TmpSignalConversionAtSFunctio_k = ((real_T)(ruleID + Sum9) + (Sum26 + Sum22))
 + (real_T)(mfIndex + tmp_6);
/* MATLAB Function: '<S17>/Evaluate Rule Antecedents' incorporates:
* DataTypeConversion: '<S17>/InputConversion'
*/
Sum 26 = 0.0;
```

 $tmp_5[0] = -13.5;$  $tmp_5[1] = -9.0;$  $tmp_5[2] = -4.5;$ inputMFCache[0] = Code\_MNLR\_v3\_trimf(TmpSignalConversionAtSFunctio\_k, tmp\_5); tmp 5[0] = -9.0; $tmp_5[1] = -4.5;$ tmp 5[2] = 0.0;inputMFCache[1] = Code\_MNLR\_v3\_trimf(TmpSignalConversionAtSFunctio\_k, tmp\_5); inputMFCache[2] = Code\_MNLR\_v3\_trimf(TmpSignalConversionAtSFunctio\_k, c); tmp 5[0] = 0.0; $tmp_5[1] = 4.5;$ tmp 5[2] = 9.0;inputMFCache[3] = Code MNLR v3 trimf(TmpSignalConversionAtSFunctio k, tmp 5);  $tmp_5[0] = 4.5;$  $tmp_5[1] = 9.0;$  $tmp_5[2] = 13.5;$ inputMFCache[4] = Code\_MNLR\_v3\_trimf(TmpSignalConversionAtSFunctio\_k, tmp\_5); tmp 5[0] = -9.0; $tmp_5[1] = -6.0;$ tmp 5[2] = -3.0: inputMFCache[5] = Code MNLR v3 trimf(0.0, tmp 5);  $tmp_5[0] = -6.0;$ tmp 5[1] = -3.0; $tmp_5[2] = 0.0;$ inputMFCache[6] = Code\_MNLR\_v3\_trimf(0.0, tmp\_5);  $tmp_5[0] = -3.0;$  $tmp_5[1] = 0.0;$ tmp 5[2] = 3.0;inputMFCache[7] = Code\_MNLR\_v3\_trimf(0.0, tmp\_5);  $tmp_5[0] = 0.0;$  $tmp_5[1] = 3.0;$  $tmp_5[2] = 6.0;$ inputMFCache[8] = Code\_MNLR\_v3\_trimf(0.0, tmp\_5);  $tmp_5[0] = 3.0;$  $tmp_5[1] = 6.0;$ tmp 5[2] = 9.0;inputMFCache[9] = Code\_MNLR\_v3\_trimf(0.0, tmp\_5); for (ruleID = 0; ruleID < 25; ruleID++) { Sum22 = inputMFCache[b 0[ruleID + 25] + 4] \* inputMFCache[b 0[ruleID] - 1];Sum 26 += Sum 22;rtb\_antecedentOutputs\_c[ruleID] = Sum22; /\* MATLAB Function: '<S17>/Evaluate Rule Consequents' \*/ Code MN EvaluateRuleConsequents(rtb antecedentOutputs c, rtb aggregatedOutputs a); /\* MATLAB Function: '<S17>/Defuzzify Outputs' incorporates: \* MATLAB Function: '<S17>/Evaluate Rule Antecedents' \*/ if (Sum 26 == 0.0) { TmpSignalConversionAtSFunctio\_k = 0.0; } else { TmpSignalConversionAtSFunctio k = 1.0 / Sum26 \* rtb aggregatedOutputs a[0]; /\* End of Outputs for SubSystem: '<Root>/Fuzzy Logic Controller1' \*/ /\* Sum: '<Root>/Sum23' \*/ rtb\_defuzzifiedOutputs\_ol = TmpSignalConversionAtSFunctio\_k + InputConversion o idx 2; /\* Outputs for Atomic SubSystem: '<Root>/Fuzzy Logic Controller1' \*/

/\* MATLAB Function: '<S17>/Defuzzify Outputs' incorporates: \* MATLAB Function: '<S17>/Evaluate Rule Antecedents' \*/ if (Sum 26 == 0.0) { TmpSignalConversionAtSFunctio k = 0.0; } else { TmpSignalConversionAtSFunctio k = 1.0 / Sum26 \* rtb aggregatedOutputs a[1]; } /\* End of Outputs for SubSystem: '<Root>/Fuzzy Logic Controller1' \*/ /\* Sum: '<Root>/Sum23' \*/ Sum6 = TmpSignalConversionAtSFunctio\_k + InputConversion\_o\_idx\_2; /\* Outputs for Atomic SubSystem: '<Root>/Fuzzy Logic Controller1' \*/ /\* MATLAB Function: '<S17>/Defuzzify Outputs' incorporates: \* MATLAB Function: '<S17>/Evaluate Rule Antecedents' \*/ if (Sum 26 == 0.0) { TmpSignalConversionAtSFunctio\_k = 0.0; } else { TmpSignalConversionAtSFunctio\_ $k = 1.0 / Sum26 * rtb_aggregatedOutputs_a[2];$ } /\* End of Outputs for SubSystem: '<Root>/Fuzzy Logic Controller1' \*/ /\* Sum: '<Root>/Sum23' \*/ Sum22 = TmpSignalConversionAtSFunctio k + InputConversion o idx 2;/\* Outputs for Atomic SubSystem: '<Root>/Fuzzy Logic Controller1' \*/ /\* MATLAB Function: '<S17>/Defuzzify Outputs' incorporates: \* MATLAB Function: '<S17>/Evaluate Rule Antecedents' \*/ if (Sum 26 == 0.0) { TmpSignalConversionAtSFunctio\_k = 0.0; } else { TmpSignalConversionAtSFunctio k = 1.0 / Sum26 \* rtb aggregatedOutputs a[3]; /\* End of Outputs for SubSystem: '<Root>/Fuzzy Logic Controller1' \*/ /\* TransportDelay: '<Root>/Transport Delay5' \*/ real\_T \*\*uBuffer = (real T\*\*) &Code MNLR v3 DW.TransportDelay5 PWORK.TUbufferPtrs[0]; real\_T simTime = Code\_MNLR\_v3\_M->Timing.t[0]; real T tMinusDelay = simTime - 0.001; Code\_MNLR\_v3\_B.TransportDelay5 = rt\_TDelayInterpolate( tMinusDelay, 0.0, \*uBuffer, Code MNLR v3 DW.TransportDelay5 IWORK.CircularBufSize, &Code MNLR v3 DW.TransportDelay5 IWORK.Last, Code\_MNLR\_v3\_DW.TransportDelay5\_IWORK.Tail, Code MNLR v3 DW.TransportDelay5 IWORK.Head, 0.0. 0, 0); } /\* Sum: '<Root>/Sum26' incorporates: \* Constant: '<Root>/Constant1' \*/ Sum 26 = (0.25 - rtb defuzzifiedOutputs ol) - Code MNLR v3 B.TransportDelay5;/\* Gain: '<S745>/Filter Coefficient' incorporates: \* Gain: '<S736>/Derivative Gain'

\* Integrator: '<S737>/Filter'

\* Sum: '<S737>/SumD'

\*/ Code\_MNLR\_v3\_B.FilterCoefficient\_i2 = (1.56E-5 \* Sum26 -Code\_MNLR\_v3\_X.Filter\_CSTATE\_d) \* 100.0; /\* SimscapeInputBlock: '<S954>/INPUT 3 1 1' incorporates: \* Gain: '<S747>/Proportional Gain' \* Integrator: '<S742>/Integrator' \* Sum: '<S751>/Sum' \*/ Code\_MNLR\_v3\_B.INPUT\_3\_1\_1[0] = (0.016 \* Sum26 + Code\_MNLR\_v3\_X.Integrator\_CSTATE\_jt) + Code\_MNLR\_v3\_B.FilterCoefficient\_i2; Code MNLR v3 B.INPUT 3 1 1[1] = 0.0;Code MNLR v3 B.INPUT 3 1 1[2] = 0.0;Code\_MNLR\_v3\_B.INPUT\_3\_1\_1[3] = 0.0; /\* TransportDelay: '<Root>/Transport Delay6' \*/ { real T \*\*uBuffer = (real T\*\*)&Code MNLR v3 DW.TransportDelay6 PWORK.TUbufferPtrs[0]; real\_T simTime = Code\_MNLR\_v3\_M->Timing.t[0]; real T tMinusDelay = simTime - 0.001; Code MNLR v3 B.TransportDelay6 = rt TDelayInterpolate( tMinusDelay, 0.0, \*uBuffer. Code\_MNLR\_v3\_DW.TransportDelay6\_IWORK.CircularBufSize, &Code MNLR v3 DW.TransportDelay6 IWORK.Last, Code\_MNLR\_v3\_DW.TransportDelay6\_IWORK.Tail, Code MNLR v3 DW.TransportDelay6 IWORK.Head, 0.0. 0. 0);/\* Sum: '<Root>/Sum6' incorporates: \* Constant: '<Root>/Constant1' \*/ Sum6 = (0.25 - Sum6) - Code MNLR v3 B.TransportDelay6;/\* Gain: '<S697>/Filter Coefficient' incorporates: \* Gain: '<S688>/Derivative Gain' \* Integrator: '<S689>/Filter' \* Sum: '<S689>/SumD' \*/ Code\_MNLR\_v3\_B.FilterCoefficient\_c = (1.56E-5 \* Sum6 -Code\_MNLR\_v3\_X.Filter\_CSTATE\_i4) \* 100.0; /\* SimscapeInputBlock: '<S954>/INPUT 4 1 1' incorporates: \* Gain: '<S699>/Proportional Gain' \* Integrator: '<S694>/Integrator' \* Sum: '<S703>/Sum' \*/  $Code_MNLR_v3_B.INPUT_4_1_1[0] = (0.016 * Sum6 +$ Code\_MNLR\_v3\_X.Integrator\_CSTATE\_kg) + Code\_MNLR\_v3\_B.FilterCoefficient\_c; Code\_MNLR\_v3\_B.INPUT\_4\_1\_1[1] = 0.0; Code MNLR v3 B.INPUT 4 1 1[2] = 0.0;Code\_MNLR\_v3\_B.INPUT\_4\_1\_1[3] = 0.0; /\* SimscapeInputBlock: '<S954>/INPUT 5 1 1' incorporates: \* Gain: '<Root>/Gain1' \*/ Code\_MNLR\_v3\_B.INPUT\_5\_1\_1[0] = -Sum5; Code\_MNLR\_v3\_B.INPUT\_5\_1\_1[1] = 0.0;

Code\_MNLR\_v3\_B.INPUT\_5\_1\_1[2] = 0.0; Code MNLR v3 B.INPUT 5 1 1[3] = 0.0;/\* TransportDelay: '<Root>/Transport Delay7' \*/ { real T \*\*uBuffer = (real T\*\*)&Code\_MNLR\_v3\_DW.TransportDelay7\_PWORK.TUbufferPtrs[0]; real T simTime = Code MNLR v3 M->Timing.t[0]; real\_T tMinusDelay = simTime - 0.001; Code\_MNLR\_v3\_B.TransportDelay7 = rt\_TDelayInterpolate( tMinusDelay, 0.0, \*uBuffer. Code MNLR v3 DW.TransportDelay7 IWORK.CircularBufSize, &Code\_MNLR\_v3\_DW.TransportDelay7\_IWORK.Last, Code\_MNLR\_v3\_DW.TransportDelay7\_IWORK.Tail, Code\_MNLR\_v3\_DW.TransportDelay7\_IWORK.Head, 0.0, 0, 0); ł /\* Sum: '<Root>/Sum22' incorporates: \* Constant: '<Root>/Constant1' \*/ Sum22 = (0.25 - Sum22) - Code\_MNLR\_v3\_B.TransportDelay7; /\* Gain: '<S793>/Filter Coefficient' incorporates: \* Gain: '<S784>/Derivative Gain' \* Integrator: '<S785>/Filter' \* Sum: '<S785>/SumD' \*/ Code\_MNLR\_v3\_B.FilterCoefficient\_c0 = (1.56E-5 \* Sum22 -Code MNLR v3 X.Filter CSTATE j) \* 100.0; /\* SimscapeInputBlock: '<S954>/INPUT\_6\_1\_1' incorporates: \* Gain: '<S795>/Proportional Gain' \* Integrator: '<S790>/Integrator' \* Sum: '<S799>/Sum' \*/  $Code_MNLR_v3_B.INPUT_6_1_1[0] = (0.016 * Sum 22 +$ Code MNLR v3 X.Integrator CSTATE j2) + Code MNLR v3 B.FilterCoefficient c0; Code MNLR v3 B.INPUT 6 1 1[1] = 0.0;Code\_MNLR\_v3\_B.INPUT\_6\_1\_1[2] = 0.0; Code\_MNLR\_v3\_B.INPUT\_6\_1\_1[3] = 0.0; /\* SimscapeInputBlock: '<S954>/INPUT\_7\_1\_1' incorporates: \* Gain: '<Root>/Gain1' \*/ Code  $MNLR_v3_B.INPUT_7_1_1[0] = -Sum5;$ Code\_MNLR\_v3\_B.INPUT\_7\_1\_1[1] = 0.0; Code\_MNLR\_v3\_B.INPUT\_7\_1\_1[2] = 0.0; Code\_MNLR\_v3\_B.INPUT\_7\_1\_1[3] = 0.0; /\* TransportDelay: '<Root>/Transport Delay8' \*/ { real\_T \*\*uBuffer = (real\_T\*\*) &Code MNLR v3 DW.TransportDelay8 PWORK.TUbufferPtrs[0]; real T simTime = Code MNLR v3 M->Timing.t[0]; real T tMinusDelay = simTime - 0.001; Code MNLR v3 B.TransportDelay8 = rt TDelayInterpolate( tMinusDelay, 0.0, \*uBuffer,

```
Code_MNLR_v3_DW.TransportDelay8_IWORK.CircularBufSize,
  &Code MNLR v3 DW.TransportDelay8 IWORK.Last,
  Code_MNLR_v3_DW.TransportDelay8_IWORK.Tail,
  Code_MNLR_v3_DW.TransportDelay8_IWORK.Head,
  0.0.
 0.
 0);
}
/* Sum: '<Root>/Sum25' incorporates:
* Constant: '<Root>/Constant1'
* Sum: '<Root>/Sum23'
*/
InputConversion o idx 2 = (0.25 - (TmpSignalConversionAtSFunctio k +
 InputConversion_o_idx_2)) - Code_MNLR_v3_B.TransportDelay8;
/* Gain: '<S841>/Filter Coefficient' incorporates:
* Gain: '<S832>/Derivative Gain'
* Integrator: '<S833>/Filter'
* Sum: '<S833>/SumD'
*/
Code MNLR v3 B.FilterCoefficient al = (1.56E-5 * InputConversion o idx 2 - 
Code_MNLR_v3_X.Filter_CSTATE_ku) * 100.0;
/* SimscapeInputBlock: '<S954>/INPUT_8_1_1' incorporates:
* Gain: '<S843>/Proportional Gain'
* Integrator: '<S838>/Integrator'
* Sum: '<S847>/Sum'
*/
Code_MNLR_v3_B.INPUT_8_1_1[0] = (0.016 * InputConversion_o_idx_2 +
 Code MNLR v3 X.Integrator CSTATE ft) + Code MNLR v3 B.FilterCoefficient al;
Code MNLR v3 B.INPUT_8_1_1[1] = 0.0;
Code_MNLR_v3_B.INPUT_8_1_1[2] = 0.0;
Code MNLR v3 B.INPUT 8 1 1[3] = 0.0;
/* SimscapeInputBlock: '<S954>/INPUT_9_1_1' */
Code_MNLR_v3_B.INPUT_9_1_1[0] = mVal;
Code MNLR v3 B.INPUT 9 1 1[1] = 0.0;
Code_MNLR_v3_B.INPUT_9_1_1[2] = 0.0;
Code MNLR v3 B.INPUT 9 1 1[3] = 0.0;
/* SimscapeExecutionBlock: '<S954>/STATE_1' incorporates:
* SimscapeExecutionBlock: '<S954>/OUTPUT 1 0'
*/
simulationData = (NeslSimulationData *)Code_MNLR_v3_DW.STATE_1_SimData;
time = Y;
simulationData->mData->mTime.mN = 1;
simulationData->mData->mTime.mX = &time;
simulationData->mData->mContStates.mN = 245;
simulationData->mData->mContStates.mX = 
 &Code_MNLR_v3_X.Code_MNLR_v3Double_Acting_Hydra[0];
simulationData->mDiscStates.mN = 0;
simulationData->mData->mDiscStates.mX = &Code_MNLR_v3_DW.STATE_1_Discrete;
simulationData->mData->mModeVector.mN = 194;
simulationData->mData->mModeVector.mX = &Code_MNLR_v3_DW.STATE_1_Modes[0];
ok = false;
simulationData->mData->mFoundZcEvents = ok;
ok = rtmIsMajorTimeStep(Code MNLR v3 M);
simulationData->mData->mIsMajorTimeStep = ok;
tmp 0 = false;
simulationData->mData->mIsSolverAssertCheck = tmp_0;
simulationData->mData->mIsSolverCheckingCIC = false;
tmp_0 = rtsiIsSolverComputingJacobian(&Code_MNLR_v3_M->solverInfo);
```

simulationData->mData->mIsComputingJacobian = tmp\_0; simulationData->mData->mIsEvaluatingF0 = false; simulationData->mData->mIsSolverRequestingReset = false; tmp\_0 = rtsiIsModeUpdateTimeStep(&Code\_MNLR\_v3\_M->solverInfo); simulationData->mData->mIsModeUpdateTimeStep = tmp 0;  $tmp_2[0] = 0;$  $tmp \ 1[0] = Code \ MNLR \ v3 \ B.INPUT \ 12 \ 1 \ 1[0];$  $tmp_1[1] = Code_MNLR_v3_B.INPUT_12_1_1[1];$ tmp\_1[2] = Code\_MNLR\_v3\_B.INPUT\_12\_1\_1[2];  $tmp_1[3] = Code_MNLR_v3_B.INPUT_12_1_1[3];$  $tmp_2[1] = 4;$ tmp 1[4] = Code MNLR v3 B.INPUT 1 1 1[0];tmp 1[5] = Code MNLR v3 B.INPUT 1 1 1[1]; $tmp_1[6] = Code_MNLR_v3_B.INPUT_1_1_1[2];$  $tmp_1[7] = Code_MNLR_v3_B.INPUT_1_1_1[3];$  $tmp_2[2] = 8;$  $tmp_1[8] = Code_MNLR_v3_B.INPUT_13_1_1[0];$  $tmp_1[9] = Code_MNLR_v3_B.INPUT_13_1_1[1];$  $tmp_1[10] = Code_MNLR_v3_B.INPUT_13_1_1[2];$ tmp 1[11] = Code MNLR v3 B.INPUT 13 1 1[3]; $tmp_2[3] = 12;$  $tmp_1[12] = Code_MNLR_v3_B.INPUT_15_1_1[0];$ tmp 1[13] = Code MNLR v3 B.INPUT 15 1 1[1]; $tmp_1[14] = Code_MNLR_v3_B.INPUT_15_1_1[2];$  $tmp_1[15] = Code_MNLR_v3_B.INPUT_15_1_1[3];$  $tmp_2[4] = 16;$ tmp\_1[16] = Code\_MNLR\_v3\_B.INPUT\_21\_1\_1[0]; tmp 1[17] = Code MNLR v3 B.INPUT 21 1 1[1];tmp  $1[18] = Code_MNLR_v3_B.INPUT_21_1_1[2];$ tmp\_1[19] = Code\_MNLR\_v3\_B.INPUT\_21\_1\_1[3]; tmp 2[5] = 20;tmp\_1[20] = Code\_MNLR\_v3\_B.INPUT\_24\_1\_1[0];  $tmp_1[21] = Code_MNLR_v3_B.INPUT_24_1_1[1];$  $tmp_1[22] = Code_MNLR_v3_B.INPUT_24_1_1[2];$ tmp\_1[23] = Code\_MNLR\_v3\_B.INPUT\_24\_1\_1[3]; tmp 2[6] = 24; $tmp_1[24] = Code_MNLR_v3_B.INPUT_18_1_1[0];$ tmp\_1[25] = Code\_MNLR\_v3\_B.INPUT\_18\_1\_1[1];  $tmp \ 1[26] = Code \ MNLR \ v3 \ B.INPUT \ 18 \ 1 \ 1[2];$ tmp\_1[27] = Code\_MNLR\_v3\_B.INPUT\_18\_1\_1[3];  $tmp_2[7] = 28;$  $tmp_1[28] = Code_MNLR_v3_B.INPUT_26_1_1[0];$ tmp\_1[29] = Code\_MNLR\_v3\_B.INPUT\_26\_1\_1[1];  $tmp \ 1[30] = Code \ MNLR \ v3 \ B.INPUT \ 26 \ 1 \ 1[2];$  $tmp_1[31] = Code_MNLR_v3_B.INPUT_26_1_1[3];$  $tmp_2[8] = 32;$ tmp 1[32] = Code MNLR v3 B.INPUT 29 1 1[0];tmp\_1[33] = Code\_MNLR\_v3\_B.INPUT\_29\_1\_1[1];  $tmp_1[34] = Code_MNLR_v3_B.INPUT_29_1_1[2];$ tmp\_1[35] = Code\_MNLR\_v3\_B.INPUT\_29\_1\_1[3];  $tmp_2[9] = 36;$ tmp 1[36] = Code MNLR v3 B.INPUT 31 1 1[0];tmp\_1[37] = Code\_MNLR\_v3\_B.INPUT\_31\_1\_1[1]; tmp\_1[38] = Code\_MNLR\_v3\_B.INPUT\_31\_1\_1[2];  $tmp_1[39] = Code_MNLR_v3_B.INPUT_31_1_1[3];$  $tmp_2[10] = 40;$  $tmp_1[40] = Code_MNLR_v3_B.INPUT_2_1_1[0];$  $tmp_1[41] = Code_MNLR_v3_B.INPUT_2_1_1[1];$ 

 $tmp_1[42] = Code_MNLR_v3_B.INPUT_2_1_1[2];$  $tmp_1[43] = Code_MNLR_v3_B.INPUT_2_1_1[3];$  $tmp_2[11] = 44;$  $tmp_1[44] = Code_MNLR_v3_B.INPUT_28_1_1[0];$ tmp\_1[45] = Code\_MNLR\_v3\_B.INPUT\_28\_1\_1[1];  $tmp_1[46] = Code_MNLR_v3_B.INPUT_28_1_1[2];$  $tmp_1[47] = Code_MNLR_v3_B.INPUT_28_1_1[3];$  $tmp_2[12] = 48;$  $tmp_1[48] = Code_MNLR_v3_B.INPUT_5_1_1[0];$  $tmp_1[49] = Code_MNLR_v3_B.INPUT_5_1_1[1];$  $tmp_1[50] = Code_MNLR_v3_B.INPUT_5_1_1[2];$ tmp 1[51] = Code MNLR v3 B.INPUT 5 1 1[3];tmp 2[13] = 52; $tmp_1[52] = Code_MNLR_v3_B.INPUT_7_1_1[0];$ tmp\_1[53] = Code\_MNLR\_v3\_B.INPUT\_7\_1\_1[1];  $tmp_1[54] = Code_MNLR_v3_B.INPUT_7_1_1[2];$  $tmp_1[55] = Code_MNLR_v3_B.INPUT_7_1_1[3];$  $tmp_2[14] = 56;$  $tmp_1[56] = Code_MNLR_v3_B.INPUT_9_1_1[0];$ tmp 1[57] = Code MNLR v3 B.INPUT 9 1 1[1]; $tmp_1[58] = Code_MNLR_v3_B.INPUT_9_1_1[2];$  $tmp_1[59] = Code_MNLR_v3_B.INPUT_9_1_1[3];$ tmp 2[15] = 60; $tmp_1[60] = Code_MNLR_v3_B.INPUT_17_1_1[0];$  $tmp_1[61] = Code_MNLR_v3_B.INPUT_17_1_1[1];$  $tmp_1[62] = Code_MNLR_v3_B.INPUT_17_1_1[2];$ tmp\_1[63] = Code\_MNLR\_v3\_B.INPUT\_17\_1\_1[3]; tmp 2[16] = 64; $tmp_1[64] = Code_MNLR_v3_B.INPUT_3_1_1[0];$  $tmp_1[65] = Code_MNLR_v3_B.INPUT_3_1_1[1];$  $tmp_1[66] = Code_MNLR_v3_B.INPUT_3_1_1[2];$  $tmp_1[67] = Code_MNLR_v3_B.INPUT_3_1_1[3];$ tmp 2[17] = 68; $tmp \ 1[68] = Code \ MNLR \ v3 \ B.INPUT \ 4 \ 1 \ 1[0];$ tmp\_1[69] = Code\_MNLR\_v3\_B.INPUT\_4\_1\_1[1]; tmp 1[70] = Code MNLR v3 B.INPUT 4 1 1[2]; $tmp_1[71] = Code_MNLR_v3_B.INPUT_4_1_1[3];$ tmp 2[18] = 72;tmp 1[72] = Code MNLR v3 B.INPUT 6 1 1[0]; $tmp_1[73] = Code_MNLR_v3_B.INPUT_6_1_1[1];$  $tmp_1[74] = Code_MNLR_v3_B.INPUT_6_1_1[2];$  $tmp_1[75] = Code_MNLR_v3_B.INPUT_6_1_1[3];$  $tmp_2[19] = 76;$  $tmp \ 1[76] = Code \ MNLR \ v3 \ B.INPUT \ 8 \ 1 \ 1[0];$  $tmp_1[77] = Code_MNLR_v3_B.INPUT_8_1_1[1];$  $tmp_1[78] = Code_MNLR_v3_B.INPUT_8_1_1[2];$  $tmp_1[79] = Code_MNLR_v3_B.INPUT_8_1_1[3];$  $tmp_2[20] = 80;$  $tmp_1[80] = Code_MNLR_v3_B.INPUT_10_1_1[0];$  $tmp_1[81] = Code_MNLR_v3_B.INPUT_10_1_1[1];$ tmp\_1[82] = Code\_MNLR\_v3\_B.INPUT\_10\_1\_1[2]; tmp 1[83] = Code MNLR v3 B.INPUT 10 1 1[3]; $tmp_2[21] = 84;$  $tmp_1[84] = Code_MNLR_v3_B.INPUT_11_1_1[0];$  $tmp \ 1[85] = Code \ MNLR \ v3 \ B.INPUT \ 11 \ 1 \ 1[1];$  $tmp_1[86] = Code_MNLR_v3_B.INPUT_11_1_1[2];$ tmp\_1[87] = Code\_MNLR\_v3\_B.INPUT\_11\_1\_1[3];  $tmp_2[22] = 88;$ 

 $tmp_1[88] = Code_MNLR_v3_B.INPUT_14_1_1[0];$ tmp\_1[89] = Code\_MNLR\_v3\_B.INPUT\_14\_1\_1[1];  $tmp_1[90] = Code_MNLR_v3_B.INPUT_14_1_1[2];$  $tmp_1[91] = Code_MNLR_v3_B.INPUT_14_1_1[3];$  $tmp_2[23] = 92;$  $tmp_1[92] = Code_MNLR_v3_B.INPUT_16_1_1[0];$ tmp 1[93] = Code MNLR v3 B.INPUT 16 1 1[1]; $tmp_1[94] = Code_MNLR_v3_B.INPUT_16_1_1[2];$  $tmp_1[95] = Code_MNLR_v3_B.INPUT_16_1_1[3];$  $tmp_2[24] = 96;$  $tmp_1[96] = Code_MNLR_v3_B.INPUT_19_1_1[0];$ tmp 1[97] = Code MNLR v3 B.INPUT 19 1 1[1];tmp 1[98] = Code MNLR v3 B.INPUT 19 1 1[2]; tmp\_1[99] = Code\_MNLR\_v3\_B.INPUT\_19\_1\_1[3]; tmp 2[25] = 100; $tmp_1[100] = Code_MNLR_v3_B.INPUT_20_1_1[0];$  $tmp_1[101] = Code_MNLR_v3_B.INPUT_20_1_1[1];$ tmp\_1[102] = Code\_MNLR\_v3\_B.INPUT\_20\_1\_1[2]; tmp\_1[103] = Code\_MNLR\_v3\_B.INPUT\_20\_1\_1[3]; tmp 2[26] = 104; $tmp_1[104] = Code_MNLR_v3_B.INPUT_22_1_1[0];$  $tmp_1[105] = Code_MNLR_v3_B.INPUT_22_1_1[1];$ tmp 1[106] = Code MNLR v3 B.INPUT 22 1 1[2]; $tmp_1[107] = Code_MNLR_v3_B.INPUT_22_1_1[3];$  $tmp_2[27] = 108;$ tmp\_1[108] = Code\_MNLR\_v3\_B.INPUT\_25\_1\_1[0]; tmp\_1[109] = Code\_MNLR\_v3\_B.INPUT\_25\_1\_1[1]; tmp 1[110] = Code MNLR v3 B.INPUT 25 1 1[2]; $tmp_1[111] = Code_MNLR_v3_B.INPUT_25_1_1[3];$  $tmp_2[28] = 112;$ tmp\_1[112] = Code\_MNLR\_v3\_B.INPUT\_23\_1\_1[0]; tmp\_1[113] = Code\_MNLR\_v3\_B.INPUT\_23\_1\_1[1];  $tmp_1[114] = Code_MNLR_v3_B.INPUT_23_1_1[2];$  $tmp_1[115] = Code_MNLR_v3_B.INPUT_23_1_1[3];$ tmp\_2[29] = 116; tmp 1[116] = Code MNLR v3 B.INPUT 27 1 1[0];  $tmp_1[117] = Code_MNLR_v3_B.INPUT_27_1_1[1];$ tmp\_1[118] = Code\_MNLR\_v3\_B.INPUT\_27\_1\_1[2];  $tmp_1[119] = Code_MNLR_v3_B.INPUT_27_1_1[3];$  $tmp_2[30] = 120;$ tmp\_1[120] = Code\_MNLR\_v3\_B.INPUT\_30\_1\_1[0]; tmp\_1[121] = Code\_MNLR\_v3\_B.INPUT\_30\_1\_1[1]; tmp\_1[122] = Code\_MNLR\_v3\_B.INPUT\_30\_1\_1[2]; tmp 1[123] = Code MNLR v3 B.INPUT 30 1 1[3]; $tmp_2[31] = 124;$ tmp\_1[124] = Code\_MNLR\_v3\_B.INPUT\_32\_1\_1[0];  $tmp_1[125] = Code_MNLR_v3_B.INPUT_32_1_1[1];$  $tmp_1[126] = Code_MNLR_v3_B.INPUT_32_1_1[2];$  $tmp_1[127] = Code_MNLR_v3_B.INPUT_32_1_1[3];$  $tmp_2[32] = 128;$ tmp\_1[128] = Code\_MNLR\_v3\_B.INPUT\_33\_1\_1[0]; tmp 1[129] = Code MNLR v3 B.INPUT 33 1 1[1];  $tmp_1[130] = Code_MNLR_v3_B.INPUT_33_1_1[2];$ tmp\_1[131] = Code\_MNLR\_v3\_B.INPUT\_33\_1\_1[3]; tmp 2[33] = 132;simulationData->mData->mInputValues.mN = 132; simulationData->mData->mInputValues.mX = &tmp\_1[0]; simulationData->mData->mInputOffsets.mN = 34;

simulationData->mData->mInputOffsets.mX = &tmp\_2[0]; simulationData->mData->mOutputs.mN = 439; simulationData->mData->mOutputs.mX = &Code\_MNLR\_v3\_B.STATE\_1[0]; simulationData->mData->mTolerances.mN = 0; simulationData->mData->mTolerances.mX = NULL: simulationData->mData->mCstateHasChanged = false; time 0 =Sum fa tmp; simulationData->mData->mTime.mN = 1; simulationData->mData->mTime.mX = &time\_0; simulationData->mData->mSampleHits.mN = 0; simulationData->mData->mSampleHits.mX = NULL; simulationData->mData->mIsFundamentalSampleHit = false; diag = (NeuDiagnosticManager \*)Code MNLR v3 DW.STATE 1 DiagMgr; diagnosticTree = neu\_diagnostic\_manager\_get\_initial\_tree(diag); ruleID = ne simulator method((NeslSimulator \*) Code\_MNLR\_v3\_DW.STATE\_1\_Simulator, NESL\_SIM\_OUTPUTS, simulationData, diag); if (ruleID != 0) { tmp 3 = error buffer is empty(rtmGetErrorStatus(Code MNLR v3 M)); if (tmp\_3) { msg 0 = rtw diagnostics msg(diagnosticTree); rtmSetErrorStatus(Code MNLR v3 M, msg 0); ł /\* SimscapeExecutionBlock: '<S954>/OUTPUT 1 0' \*/ simulationData = (NeslSimulationData \*)Code\_MNLR\_v3\_DW.OUTPUT\_1\_0\_SimData; time\_1 = Y; simulationData->mData->mTime.mN = 1; simulationData->mData->mTime.mX = &time\_1; simulationData->mData->mContStates.mN = 0; simulationData->mData->mContStates.mX = NULL: simulationData->mDiscStates.mN = 0; simulationData->mData->mDiscStates.mX = &Code MNLR v3 DW.OUTPUT 1 0 Discrete; simulationData->mData->mModeVector.mN = 0; simulationData->mData->mModeVector.mX = &Code MNLR v3 DW.OUTPUT 1 0 Modes; tmp 3 =false; simulationData->mData->mFoundZcEvents = tmp\_3; simulationData->mData->mIsMajorTimeStep = ok; ok = false:simulationData->mData->mIsSolverAssertCheck = ok; simulationData->mData->mIsSolverCheckingCIC = false; simulationData->mData->mIsComputingJacobian = false; simulationData->mData->mIsEvaluatingF0 = false; simulationData->mIsSolverRequestingReset = false; simulationData->mData->mIsModeUpdateTimeStep = tmp\_0; tmp 4[0] = 0;Code MNLR v3 B.dv[0] = Code MNLR v3 B.INPUT 12 1 1[0];  $Code_MNLR_v3_B.dv[1] = Code_MNLR_v3_B.INPUT_12_1_1[1];$ Code MNLR v3 B.dv[2] = Code MNLR v3 B.INPUT 12 1 1[2];  $Code_MNLR_v3_B.dv[3] = Code_MNLR_v3_B.INPUT_12_1_1[3];$ tmp 4[1] = 4; $Code_MNLR_v3_B.dv[4] = Code_MNLR_v3_B.INPUT_1_1_1[0];$ Code\_MNLR\_v3\_B.dv[5] = Code\_MNLR\_v3\_B.INPUT\_1\_1\_1[1];  $Code_MNLR_v3_B.dv[6] = Code_MNLR_v3_B.INPUT_1_1_1[2];$  $Code_MNLR_v3_B.dv[7] = Code_MNLR_v3_B.INPUT_1_1_1[3];$ tmp 4[2] = 8;Code MNLR v3 B.dv[8] = Code MNLR v3 B.INPUT 13 1 1[0];  $Code_MNLR_v3_B.dv[9] = Code_MNLR_v3_B.INPUT_13_1_1[1];$ Code\_MNLR\_v3\_B.dv[10] = Code\_MNLR\_v3\_B.INPUT\_13\_1\_1[2];  $Code_MNLR_v3_B.dv[11] = Code_MNLR_v3_B.INPUT_13_1_1[3];$ 

 $tmp_4[3] = 12;$ Code\_MNLR\_v3\_B.dv[12] = Code\_MNLR\_v3\_B.INPUT\_15\_1\_1[0];  $Code_MNLR_v3_B.dv[13] = Code_MNLR_v3_B.INPUT_15_1_1[1];$ Code\_MNLR\_v3\_B.dv[14] = Code\_MNLR\_v3\_B.INPUT\_15\_1\_1[2]; Code MNLR v3 B.dv[15] = Code MNLR v3 B.INPUT 15 1 1[3];  $tmp_4[4] = 16;$ Code MNLR v3 B.dv[16] = Code MNLR v3 B.INPUT 21 1 1[0];  $Code_MNLR_v3_B.dv[17] = Code_MNLR_v3_B.INPUT_21_1_1[1];$ Code\_MNLR\_v3\_B.dv[18] = Code\_MNLR\_v3\_B.INPUT\_21\_1\_1[2]; Code MNLR v3 B.dv[19] = Code MNLR v3 B.INPUT 21 1 1[3]; $tmp_4[5] = 20;$ Code MNLR v3 B.dv[20] = Code MNLR v3 B.INPUT 24 1 1[0]; Code MNLR v3 B.dv[21] = Code MNLR v3 B.INPUT 24 1 1[1]; Code\_MNLR\_v3\_B.dv[22] = Code\_MNLR\_v3\_B.INPUT\_24\_1\_1[2]; Code MNLR v3 B.dv[23] = Code MNLR v3 B.INPUT 24 1 1[3]; $tmp_4[6] = 24;$ Code\_MNLR\_v3\_B.dv[24] = Code\_MNLR\_v3\_B.INPUT\_18\_1\_1[0]; Code MNLR v3 B.dv[25] = Code MNLR v3 B.INPUT 18 1 1[1]; Code\_MNLR\_v3\_B.dv[26] = Code\_MNLR\_v3\_B.INPUT\_18\_1\_1[2]; Code MNLR v3 B.dv[27] = Code MNLR v3 B.INPUT 18 1 1[3];tmp 4[7] = 28; $Code_MNLR_v3_B.dv[28] = Code_MNLR_v3_B.INPUT_26_1_1[0];$ Code MNLR v3 B.dv[29] = Code MNLR v3 B.INPUT 26 1 1[1]; $Code_MNLR_v3_B.dv[30] = Code_MNLR_v3_B.INPUT_26_1_1[2];$ Code\_MNLR\_v3\_B.dv[31] = Code\_MNLR\_v3\_B.INPUT\_26\_1\_1[3]; tmp 4[8] = 32;Code\_MNLR\_v3\_B.dv[32] = Code\_MNLR\_v3\_B.INPUT\_29\_1\_1[0]; Code MNLR v3 B.dv[33] = Code MNLR v3 B.INPUT 29 1 1[1];Code MNLR v3 B.dv[34] = Code MNLR v3 B.INPUT 29 1 1[2];Code\_MNLR\_v3\_B.dv[35] = Code\_MNLR\_v3\_B.INPUT\_29\_1\_1[3]; tmp 4[9] = 36;Code\_MNLR\_v3\_B.dv[36] = Code\_MNLR\_v3\_B.INPUT\_31\_1\_1[0]; Code\_MNLR\_v3\_B.dv[37] = Code\_MNLR\_v3\_B.INPUT\_31\_1\_1[1]; Code MNLR v3 B.dv[38] = Code MNLR v3 B.INPUT 31 1 1[2];Code\_MNLR\_v3\_B.dv[39] = Code\_MNLR\_v3\_B.INPUT\_31\_1\_1[3]; tmp 4[10] = 40; $Code_MNLR_v3_B.dv[40] = Code_MNLR_v3_B.INPUT_2_1_1[0];$  $Code_MNLR_v3_B.dv[41] = Code_MNLR_v3_B.INPUT_2_1_1[1];$ Code MNLR v3 B.dv[42] = Code MNLR v3 B.INPUT 2 1 1[2]; $Code_MNLR_v3_B.dv[43] = Code_MNLR_v3_B.INPUT_2_1_1[3];$  $tmp_4[11] = 44;$ Code\_MNLR\_v3\_B.dv[44] = Code\_MNLR\_v3\_B.INPUT\_28\_1\_1[0]; Code\_MNLR\_v3\_B.dv[45] = Code\_MNLR\_v3\_B.INPUT\_28\_1\_1[1]; Code MNLR v3 B.dv[46] = Code MNLR v3 B.INPUT 28 1 1[2]; Code MNLR  $v3_B.dv[47] = Code_MNLR_v3_B.INPUT_28_1_1[3];$  $tmp_4[12] = 48;$ Code MNLR v3 B.dv[48] = Code MNLR v3 B.INPUT 5 1 1[0];Code\_MNLR\_v3\_B.dv[49] = Code\_MNLR\_v3\_B.INPUT\_5\_1\_1[1]; Code\_MNLR\_v3\_B.dv[50] = Code\_MNLR\_v3\_B.INPUT\_5\_1\_1[2];  $Code_MNLR_v3_B.dv[51] = Code_MNLR_v3_B.INPUT_5_1_1[3];$  $tmp_4[13] = 52;$ Code MNLR v3 B.dv[52] = Code MNLR v3 B.INPUT 7 1 1[0]; $Code_MNLR_v3_B.dv[53] = Code_MNLR_v3_B.INPUT_7_1_1[1];$ Code\_MNLR\_v3\_B.dv[54] = Code\_MNLR\_v3\_B.INPUT\_7\_1\_1[2]; Code MNLR v3 B.dv[55] = Code MNLR v3 B.INPUT 7 1 1[3]; $tmp_4[14] = 56;$ Code\_MNLR\_v3\_B.dv[56] = Code\_MNLR\_v3\_B.INPUT\_9\_1\_1[0];  $Code_MNLR_v3_B.dv[57] = Code_MNLR_v3_B.INPUT_9_1_1[1];$ 

 $Code_MNLR_v3_B.dv[58] = Code_MNLR_v3_B.INPUT_9_1_1[2];$ Code\_MNLR\_v3\_B.dv[59] = Code\_MNLR\_v3\_B.INPUT\_9\_1\_1[3];  $tmp_4[15] = 60;$  $Code_MNLR_v3_B.dv[60] = Code_MNLR_v3_B.INPUT_17_1_1[0];$ Code\_MNLR\_v3\_B.dv[61] = Code\_MNLR\_v3\_B.INPUT\_17\_1\_1[1]; Code\_MNLR\_v3\_B.dv[62] = Code\_MNLR\_v3\_B.INPUT\_17\_1\_1[2]; Code MNLR v3 B.dv[63] = Code MNLR v3 B.INPUT 17 1 1[3]; $tmp_4[16] = 64;$  $Code_MNLR_v3_B.dv[64] = Code_MNLR_v3_B.INPUT_3_1_1[0];$ Code\_MNLR\_v3\_B.dv[65] = Code\_MNLR\_v3\_B.INPUT\_3\_1\_1[1];  $Code_MNLR_v3_B.dv[66] = Code_MNLR_v3_B.INPUT_3_1_1[2];$ Code MNLR v3 B.dv[67] = Code MNLR v3 B.INPUT 3 1 1[3]; tmp 4[17] = 68; $Code_MNLR_v3_B.dv[68] = Code_MNLR_v3_B.INPUT_4_1_1[0];$ Code MNLR v3 B.dv[69] = Code MNLR v3 B.INPUT 4 1 1[1]; $Code_MNLR_v3_B.dv[70] = Code_MNLR_v3_B.INPUT_4_1_1[2];$  $Code_MNLR_v3_B.dv[71] = Code_MNLR_v3_B.INPUT_4_1_1[3];$ tmp 4[18] = 72; $Code_MNLR_v3_B.dv[72] = Code_MNLR_v3_B.INPUT_6_1_1[0];$ Code MNLR v3 B.dv[73] = Code MNLR v3 B.INPUT 6 1 1[1];Code\_MNLR\_v3\_B.dv[74] = Code\_MNLR\_v3\_B.INPUT\_6\_1\_1[2];  $Code_MNLR_v3_B.dv[75] = Code_MNLR_v3_B.INPUT_6_1_1[3];$ tmp 4[19] = 76; $Code_MNLR_v3_B.dv[76] = Code_MNLR_v3_B.INPUT_8_1_1[0];$ Code\_MNLR\_v3\_B.dv[77] = Code\_MNLR\_v3\_B.INPUT\_8\_1\_1[1]; Code MNLR v3 B.dv[78] = Code MNLR v3 B.INPUT 8 1 1[2];Code\_MNLR\_v3\_B.dv[79] = Code\_MNLR\_v3\_B.INPUT\_8\_1\_1[3]; tmp 4[20] = 80;Code MNLR v3 B.dv[80] = Code MNLR v3 B.INPUT 10 1 1[0];Code\_MNLR\_v3\_B.dv[81] = Code\_MNLR\_v3\_B.INPUT\_10\_1\_1[1]; Code MNLR v3 B.dv[82] = Code MNLR v3 B.INPUT 10 1 1[2]; $Code_MNLR_v3_B.dv[83] = Code_MNLR_v3_B.INPUT_10_1_1[3];$ tmp 4[21] = 84;Code MNLR v3 B.dv[84] = Code MNLR v3 B.INPUT 11 1 1[0];Code\_MNLR\_v3\_B.dv[85] = Code\_MNLR\_v3\_B.INPUT\_11\_1[1]; Code MNLR v3 B.dv[86] = Code MNLR v3 B.INPUT 11 1 1[2]; $Code_MNLR_v3_B.dv[87] = Code_MNLR_v3_B.INPUT_11_1_1[3];$ tmp 4[22] = 88;Code MNLR v3 B.dv[88] = Code MNLR v3 B.INPUT 14 1 1[0];Code\_MNLR\_v3\_B.dv[89] = Code\_MNLR\_v3\_B.INPUT\_14\_1\_1[1]; Code\_MNLR\_v3\_B.dv[90] = Code\_MNLR\_v3\_B.INPUT\_14\_1\_1[2];  $Code_MNLR_v3_B.dv[91] = Code_MNLR_v3_B.INPUT_14_1_1[3];$ tmp\_4[23] = 92; Code MNLR v3 B.dv[92] = Code MNLR v3 B.INPUT 16 1 1[0];Code MNLR v3 B.dv[93] = Code MNLR v3 B.INPUT 16 1 1[1]; $Code_MNLR_v3_B.dv[94] = Code_MNLR_v3_B.INPUT_16_1_1[2];$ Code MNLR v3 B.dv[95] = Code MNLR v3 B.INPUT 16 1 1[3]; tmp 4[24] = 96;  $Code_MNLR_v3_B.dv[96] = Code_MNLR_v3_B.INPUT_19_1_1[0];$  $Code_MNLR_v3_B.dv[97] = Code_MNLR_v3_B.INPUT_19_1_1[1];$ Code\_MNLR\_v3\_B.dv[98] = Code\_MNLR\_v3\_B.INPUT\_19\_1\_1[2]; Code MNLR v3 B.dv[99] = Code MNLR v3 B.INPUT 19 1 1[3];tmp 4[25] = 100;Code MNLR v3 B.dv[100] = Code MNLR v3 B.INPUT 20 1 1[0]; Code MNLR v3 B.dv[101] = Code MNLR v3 B.INPUT 20 1 1[1]; $Code_MNLR_v3_B.dv[102] = Code_MNLR_v3_B.INPUT_20_1_1[2];$ Code\_MNLR\_v3\_B.dv[103] = Code\_MNLR\_v3\_B.INPUT\_20\_1\_1[3];  $tmp_4[26] = 104;$ 

 $Code_MNLR_v3_B.dv[104] = Code_MNLR_v3_B.INPUT_22_1_1[0];$ Code MNLR v3 B.dv[105] = Code MNLR v3 B.INPUT 22 1 1[1];  $Code_MNLR_v3_B.dv[106] = Code_MNLR_v3_B.INPUT_22_1_1[2];$ Code\_MNLR\_v3\_B.dv[107] = Code\_MNLR\_v3\_B.INPUT\_22\_1\_1[3]; tmp 4[27] = 108;Code\_MNLR\_v3\_B.dv[108] = Code\_MNLR\_v3\_B.INPUT\_25\_1\_1[0]; Code MNLR v3 B.dv[109] = Code MNLR v3 B.INPUT 25 1 1[1]; $Code_MNLR_v3_B.dv[110] = Code_MNLR_v3_B.INPUT_25_1_1[2];$ Code\_MNLR\_v3\_B.dv[111] = Code\_MNLR\_v3\_B.INPUT\_25\_1\_1[3]; tmp 4[28] = 112; $Code_MNLR_v3_B.dv[112] = Code_MNLR_v3_B.INPUT_23_1_1[0];$ Code MNLR v3 B.dv[113] = Code MNLR v3 B.INPUT 23 1 1[1];Code MNLR v3 B.dv[114] = Code MNLR v3 B.INPUT 23 1 1[2]; $Code_MNLR_v3_B.dv[115] = Code_MNLR_v3_B.INPUT_23_1_1[3];$ tmp 4[29] = 116; $Code_MNLR_v3_B.dv[116] = Code_MNLR_v3_B.INPUT_27_1_1[0];$ Code\_MNLR\_v3\_B.dv[117] = Code\_MNLR\_v3\_B.INPUT\_27\_1\_1[1]; Code\_MNLR\_v3\_B.dv[118] = Code\_MNLR\_v3\_B.INPUT\_27\_1\_1[2];  $Code_MNLR_v3_B.dv[119] = Code_MNLR_v3_B.INPUT_27_1_1[3];$ tmp 4[30] = 120: Code MNLR v3 B.dv[120] = Code MNLR v3 B.INPUT 30 1 1[0];  $Code_MNLR_v3_B.dv[121] = Code_MNLR_v3_B.INPUT_30_1_1[1];$ Code MNLR v3 B.dv[122] = Code MNLR v3 B.INPUT 30 1 1[2]; $Code_MNLR_v3_B.dv[123] = Code_MNLR_v3_B.INPUT_30_1_1[3];$  $tmp_4[31] = 124;$ Code\_MNLR\_v3\_B.dv[124] = Code\_MNLR\_v3\_B.INPUT\_32\_1\_1[0]; Code\_MNLR\_v3\_B.dv[125] = Code\_MNLR\_v3\_B.INPUT\_32\_1\_1[1]; Code MNLR v3 B.dv[126] = Code MNLR v3 B.INPUT 32 1 1[2]; $Code_MNLR_v3_B.dv[127] = Code_MNLR_v3_B.INPUT_32_1_1[3];$  $tmp_4[32] = 128;$ Code MNLR v3 B.dv[128] = Code MNLR v3 B.INPUT 33 1 1[0];Code\_MNLR\_v3\_B.dv[129] = Code\_MNLR\_v3\_B.INPUT\_33\_1\_1[1]; Code\_MNLR\_v3\_B.dv[130] = Code\_MNLR\_v3\_B.INPUT\_33\_1\_1[2]; Code\_MNLR\_v3\_B.dv[131] = Code\_MNLR\_v3\_B.INPUT\_33\_1\_1[3];  $tmp_4[33] = 132;$ memcpy(&Code MNLR v3 B.dv[132], &Code MNLR v3 B.STATE 1[0], 439U \* sizeof (real\_T)); tmp 4[34] = 571;simulationData->mData->mInputValues.mN = 571; simulationData->mData->mInputValues.mX = &Code\_MNLR\_v3\_B.dv[0]; simulationData->mData->mInputOffsets.mN = 35; simulationData->mData->mInputOffsets.mX = &tmp\_4[0]; simulationData->mData->mOutputs.mN = 32; simulationData->mData->mOutputs.mX = &Code MNLR v3 B.OUTPUT 1 0[0]; simulationData->mData->mTolerances.mN = 0; simulationData->mData->mTolerances.mX = NULL; simulationData->mData->mCstateHasChanged = false; time\_2 = Sum\_fa\_tmp; simulationData->mData->mTime.mN = 1; simulationData->mData->mTime.mX = &time 2; simulationData->mData->mSampleHits.mN = 0; simulationData->mData->mSampleHits.mX = NULL; simulationData->mData->mIsFundamentalSampleHit = false; diag = (NeuDiagnosticManager \*)Code\_MNLR\_v3\_DW.OUTPUT\_1\_0\_DiagMgr; diagnosticTree 0 = neu diagnostic manager get initial tree(diag); ruleID = ne\_simulator\_method((NeslSimulator \*) Code\_MNLR\_v3\_DW.OUTPUT\_1\_0\_Simulator, NESL\_SIM\_OUTPUTS, simulationData, diag);

```
if (ruleID != 0) {
 ok = error buffer is empty(rtmGetErrorStatus(Code MNLR v3 M));
 if (ok) {
  msg_1 = rtw_diagnostics_msg(diagnosticTree_0);
  rtmSetErrorStatus(Code MNLR v3 M, msg 1);
 }
}
/* Gain: '<S835>/Integral Gain' */
Code_MNLR_v3_B.IntegralGain = 3.23E-5 * InputConversion_o_idx_2;
/* Gain: '<S787>/Integral Gain' */
Code_MNLR_v3_B.IntegralGain_l = 3.23E-5 * Sum22;
/* Gain: '<S691>/Integral Gain' */
Code MNLR v3 B.IntegralGain d = 3.23E-5 * Sum6;
/* Gain: '<S739>/Integral Gain' */
Code MNLR v3 B.IntegralGain o = 3.23E-5 * Sum26;
/* TransferFcn: '<S103>/Transfer Fcn6' */
Y = 48.999814587829867 * Code_MNLR_v3_X.TransferFcn6_CSTATE;
/* Step: '<S103>/Step1' */
if (Sum_fa_tmp < 0.0) {
 Sum 26 = 0.0;
} else {
 Sum 26 = 9.9;
ł
/* Sum: '<S103>/Sum5' incorporates:
* Gain: '<S103>/Gain5'
* Gain: '<S103>/Gain7'
* Gain: '<S103>/Gain8'
* Step: '<S103>/Step1'
* Sum: '<S103>/Sum4'
*/
Code MNLR v3 B.Sum5 e = (Sum26 - 0.052685774264903285 * Sum fa) *
 79.067569437344 - 0.23529411764705882 * Y;
/* TransferFcn: '<S103>/Transfer Fcn4' */
Code MNLR v3 B.TransferFcn4 = 0.0;
Code_MNLR_v3_B.TransferFcn4 += 6.7857142857142838 *
 Code MNLR v3 X.TransferFcn4 CSTATE;
Code_MNLR_v3_B.TransferFcn4 += 0.22528577162622157 * Code_MNLR_v3_B.Sum5_e;
/* Gain: '<S103>/Gain6' */
Code MNLR v3 B.Gain6 = 1.3112172070177803 * Y;
/* Sum: '<S103>/Sum6' incorporates:
* Gain: '<S103>/Gain9'
* TransferFcn: '<S103>/Transfer Fcn5'
*/
Code MNLR v3 B.Sum6 j = 3928.5714285714289 *
 Code_MNLR_v3_X.TransferFcn5_CSTATE - 1.3112172070177803 * Sum_fa;
/* Gain: '<S643>/Integral Gain' */
Code MNLR v3 B.IntegralGain e = 3.23E-5 * rtb defuzzifiedOutputs g idx 0;
/* Gain: '<S595>/Integral Gain' */
Code_MNLR_v3_B.IntegralGain_m = 3.23E-5 * Sum2;
/* Gain: '<S163>/Integral Gain' */
Code MNLR v3 B.IntegralGain i = 3.23E-5 * Sum 34;
/* Gain: '<S451>/Integral Gain' */
Code_MNLR_v3_B.IntegralGain_g = 3.23E-5 * rtb_defuzzifiedOutputs_g_idx_3;
/* Gain: '<S547>/Integral Gain' */
Code_MNLR_v3_B.IntegralGain_a = 3.23E-5 * Y_f;
/* Gain: '<S403>/Integral Gain' */
Code_MNLR_v3_B.IntegralGain_em = 3.23E-5 * InputConversion_o_idx_0;
```

/\* Gain: '<S499>/Integral Gain' \*/ Code MNLR v3 B.IntegralGain f = 3.23E-5 \* Sum35;/\* Gain: '<S355>/Integral Gain' \*/ Code\_MNLR\_v3\_B.IntegralGain\_o3 = 3.23E-5 \* InputConversion\_o\_idx\_1; /\* Gain: '<S307>/Integral Gain' \*/ Code\_MNLR\_v3\_B.IntegralGain\_c = 3.23E-5 \* rtb\_defuzzifiedOutputs\_o\_idx\_3; /\* Gain: '<S259>/Integral Gain' \*/ Code\_MNLR\_v3\_B.IntegralGain\_b = 3.23E-5 \* rtb\_defuzzifiedOutputs\_o\_idx\_2; /\* Gain: '<S883>/Integral Gain' \*/ Code\_MNLR\_v3\_B.IntegralGain\_k = 3.23E-5 \* rtb\_defuzzifiedOutputs\_o\_idx\_1; /\* Gain: '<S211>/Integral Gain' \*/ Code MNLR v3 B.IntegralGain en = 3.23E-5 \* rtb defuzzifiedOutputs o idx 0; if (rtmIsMajorTimeStep(Code\_MNLR\_v3\_M)) { NeslSimulationData \*simulationData; NeuDiagnosticManager \*diagnosticManager; NeuDiagnosticTree \*diagnosticTree; char \*msg; real\_T tmp\_0[132]; real T time; int32\_T tmp\_2; int\_T tmp\_1[34]; boolean T tmp; /\* Update for TransportDelay: '<Root>/Transport Delay9' \*/ { real T \*\*uBuffer = (real T\*\*)&Code\_MNLR\_v3\_DW.TransportDelay9\_PWORK.TUbufferPtrs[0]; real T simTime = Code MNLR v3 M->Timing.t[0]; Code MNLR v3 DW.TransportDelay9 IWORK.Head = ((Code\_MNLR\_v3\_DW.TransportDelay9\_IWORK.Head < (Code\_MNLR\_v3\_DW.TransportDelay9\_IWORK.CircularBufSize-1)) ? (Code\_MNLR\_v3\_DW.TransportDelay9\_IWORK.Head+1): 0); if (Code\_MNLR\_v3\_DW.TransportDelay9\_IWORK.Head == Code MNLR v3 DW.TransportDelay9 IWORK.Tail) { Code\_MNLR\_v3\_DW.TransportDelay9\_IWORK.Tail = ((Code MNLR v3 DW.TransportDelay9 IWORK.Tail < (Code\_MNLR\_v3\_DW.TransportDelay9\_IWORK.CircularBufSize-1)) ? (Code MNLR v3 DW.TransportDelay9 IWORK.Tail+1):0); } (\*uBuffer + Code\_MNLR\_v3\_DW.TransportDelay9\_IWORK.CircularBufSize) [Code\_MNLR\_v3\_DW.TransportDelay9\_IWORK.Head] = simTime; (\*uBuffer)[Code\_MNLR\_v3\_DW.TransportDelay9\_IWORK.Head] = Code\_MNLR\_v3\_B.OUTPUT\_1\_0[18]; /\* Update for TransportDelay: '<Root>/Transport Delay10' \*/ real T \*\*uBuffer = (real T\*\*)&Code\_MNLR\_v3\_DW.TransportDelay10\_PWORK.TUbufferPtrs[0]; real\_T simTime = Code\_MNLR\_v3\_M->Timing.t[0]; Code MNLR v3 DW.TransportDelay10 IWORK.Head = ((Code\_MNLR\_v3\_DW.TransportDelay10\_IWORK.Head < (Code MNLR v3 DW.TransportDelay10 IWORK.CircularBufSize-1)) ? (Code MNLR v3 DW.TransportDelay10 IWORK.Head+1): 0); if (Code MNLR v3 DW.TransportDelay10 IWORK.Head == Code MNLR v3 DW.TransportDelay10 IWORK.Tail) { Code\_MNLR\_v3\_DW.TransportDelay10\_IWORK.Tail = ((Code MNLR v3 DW.TransportDelay10 IWORK.Tail < (Code\_MNLR\_v3\_DW.TransportDelay10\_IWORK.CircularBufSize-1)) ?

(Code\_MNLR\_v3\_DW.TransportDelay10\_IWORK.Tail+1): 0); } (\*uBuffer + Code\_MNLR\_v3\_DW.TransportDelay10\_IWORK.CircularBufSize) [Code MNLR v3 DW.TransportDelay10 IWORK.Head] = simTime; (\*uBuffer)[Code MNLR v3 DW.TransportDelay10 IWORK.Head] = Code\_MNLR\_v3\_B.OUTPUT\_1\_0[31]; ł /\* Update for TransportDelay: '<Root>/Transport Delay11' \*/ real T \*\*uBuffer = (real T\*\*)&Code\_MNLR\_v3\_DW.TransportDelay11\_PWORK.TUbufferPtrs[0]; real T simTime = Code MNLR v3 M->Timing.t[0]; Code MNLR v3 DW.TransportDelay11 IWORK.Head = ((Code\_MNLR\_v3\_DW.TransportDelay11\_IWORK.Head < (Code MNLR v3 DW.TransportDelay11 IWORK.CircularBufSize-1))? (Code\_MNLR\_v3\_DW.TransportDelay11\_IWORK.Head+1): 0); if (Code MNLR v3 DW.TransportDelay11 IWORK.Head == Code MNLR v3 DW.TransportDelay11 IWORK.Tail) { Code\_MNLR\_v3\_DW.TransportDelay11\_IWORK.Tail = ((Code\_MNLR\_v3\_DW.TransportDelay11\_IWORK.Tail < (Code MNLR v3 DW.TransportDelay11 IWORK.CircularBufSize-1))? (Code\_MNLR\_v3\_DW.TransportDelay11\_IWORK.Tail+1): 0); } (\*uBuffer + Code\_MNLR\_v3\_DW.TransportDelay11\_IWORK.CircularBufSize) [Code\_MNLR\_v3\_DW.TransportDelay11\_IWORK.Head] = simTime; (\*uBuffer)[Code MNLR v3 DW.TransportDelay11 IWORK.Head] = Code\_MNLR\_v3\_B.OUTPUT\_1\_0[16]; } /\* Update for TransportDelay: '<Root>/Transport Delay12' \*/ { real T \*\*uBuffer = (real\_T\*\*) &Code\_MNLR\_v3\_DW.TransportDelay12\_PWORK.TUbufferPtrs[0]; real\_T simTime = Code\_MNLR\_v3\_M->Timing.t[0]; Code MNLR v3 DW.TransportDelay12 IWORK.Head = ((Code\_MNLR\_v3\_DW.TransportDelay12\_IWORK.Head < (Code MNLR v3 DW.TransportDelay12 IWORK.CircularBufSize-1))? (Code MNLR v3 DW.TransportDelay12 IWORK.Head+1): 0); if (Code MNLR v3 DW.TransportDelay12 IWORK.Head == Code MNLR v3 DW.TransportDelay12 IWORK.Tail) { Code\_MNLR\_v3\_DW.TransportDelay12\_IWORK.Tail = ((Code MNLR v3 DW.TransportDelay12 IWORK.Tail < (Code\_MNLR\_v3\_DW.TransportDelay12\_IWORK.CircularBufSize-1)) ? (Code\_MNLR\_v3\_DW.TransportDelay12\_IWORK.Tail+1): 0); (\*uBuffer + Code MNLR v3 DW.TransportDelay12 IWORK.CircularBufSize) [Code\_MNLR\_v3\_DW.TransportDelay12\_IWORK.Head] = simTime; (\*uBuffer)[Code MNLR v3 DW.TransportDelay12 IWORK.Head] = Code\_MNLR\_v3\_B.OUTPUT\_1\_0[17]; ł /\* Update for TransportDelay: '<Root>/Transport Delay13' \*/ { real\_T \*\*uBuffer = (real\_T\*\*) &Code\_MNLR\_v3\_DW.TransportDelay13\_PWORK.TUbufferPtrs[0]; real T simTime = Code MNLR v3 M->Timing.t[0]; Code MNLR v3 DW.TransportDelay13 IWORK.Head = ((Code\_MNLR\_v3\_DW.TransportDelay13\_IWORK.Head < (Code\_MNLR\_v3\_DW.TransportDelay13\_IWORK.CircularBufSize-1)) ? (Code\_MNLR\_v3\_DW.TransportDelay13\_IWORK.Head+1): 0);

200

if (Code\_MNLR\_v3\_DW.TransportDelay13\_IWORK.Head == Code MNLR v3 DW.TransportDelay13 IWORK.Tail) { Code\_MNLR\_v3\_DW.TransportDelay13\_IWORK.Tail =  $((Code\_MNLR\_v3\_DW.TransportDelay13\_IWORK.Tail <$ (Code MNLR v3 DW.TransportDelay13 IWORK.CircularBufSize-1))? (Code\_MNLR\_v3\_DW.TransportDelay13\_IWORK.Tail+1): 0); } (\*uBuffer + Code\_MNLR\_v3\_DW.TransportDelay13\_IWORK.CircularBufSize) [Code\_MNLR\_v3\_DW.TransportDelay13\_IWORK.Head] = simTime; (\*uBuffer)[Code MNLR v3 DW.TransportDelay13 IWORK.Head] = Code\_MNLR\_v3\_B.OUTPUT\_1\_0[21]; /\* Update for TransportDelay: '<Root>/Transport Delay14' \*/ { real T \*\*uBuffer = (real T\*\*)&Code\_MNLR\_v3\_DW.TransportDelay14\_PWORK.TUbufferPtrs[0]; real\_T simTime = Code\_MNLR\_v3\_M->Timing.t[0]; Code MNLR v3 DW.TransportDelay14 IWORK.Head = ((Code\_MNLR\_v3\_DW.TransportDelay14\_IWORK.Head < (Code MNLR v3 DW.TransportDelay14 IWORK.CircularBufSize-1))? (Code MNLR v3 DW.TransportDelay14 IWORK.Head+1): 0); if (Code\_MNLR\_v3\_DW.TransportDelay14\_IWORK.Head == Code MNLR v3 DW.TransportDelay14 IWORK.Tail) { Code\_MNLR\_v3\_DW.TransportDelay14\_IWORK.Tail = ((Code\_MNLR\_v3\_DW.TransportDelay14\_IWORK.Tail < (Code MNLR v3 DW.TransportDelay14 IWORK.CircularBufSize-1))? (Code\_MNLR\_v3\_DW.TransportDelay14\_IWORK.Tail+1): 0); } (\*uBuffer + Code MNLR v3 DW.TransportDelay14 IWORK.CircularBufSize) [Code\_MNLR\_v3\_DW.TransportDelay14\_IWORK.Head] = simTime; (\*uBuffer)[Code MNLR v3 DW.TransportDelay14 IWORK.Head] = Code\_MNLR\_v3\_B.OUTPUT\_1\_0[22]; /\* Update for TransportDelay: '<Root>/Transport Delay15' \*/ { real T \*\*uBuffer = (real T\*\*)&Code MNLR v3 DW.TransportDelay15 PWORK.TUbufferPtrs[0]; real\_T simTime = Code\_MNLR\_v3\_M->Timing.t[0]; Code MNLR v3 DW.TransportDelay15 IWORK.Head = ((Code\_MNLR\_v3\_DW.TransportDelay15\_IWORK.Head < (Code MNLR v3 DW.TransportDelay15 IWORK.CircularBufSize-1))? (Code\_MNLR\_v3\_DW.TransportDelay15\_IWORK.Head+1): 0); if (Code\_MNLR\_v3\_DW.TransportDelay15\_IWORK.Head == Code MNLR v3 DW.TransportDelay15 IWORK.Tail) { Code MNLR v3 DW.TransportDelay15 IWORK.Tail = ((Code\_MNLR\_v3\_DW.TransportDelay15\_IWORK.Tail < (Code MNLR v3 DW.TransportDelay15 IWORK.CircularBufSize-1))? (Code\_MNLR\_v3\_DW.TransportDelay15\_IWORK.Tail+1): 0); } (\*uBuffer + Code MNLR v3 DW.TransportDelay15 IWORK.CircularBufSize) [Code\_MNLR\_v3\_DW.TransportDelay15\_IWORK.Head] = simTime; (\*uBuffer)[Code MNLR v3 DW.TransportDelay15 IWORK.Head] = Code\_MNLR\_v3\_B.OUTPUT\_1\_0[19]; /\* Update for TransportDelay: '<Root>/Transport Delay1' \*/ { real T \*\*uBuffer = (real T\*\*)

&Code\_MNLR\_v3\_DW.TransportDelay1\_PWORK.TUbufferPtrs[0];

201

real\_T simTime = Code\_MNLR\_v3\_M->Timing.t[0]; Code MNLR v3 DW.TransportDelay1 IWORK.Head = ((Code\_MNLR\_v3\_DW.TransportDelay1\_IWORK.Head < (Code\_MNLR\_v3\_DW.TransportDelay1\_IWORK.CircularBufSize-1)) ? (Code MNLR v3 DW.TransportDelay1 IWORK.Head+1): 0); if (Code\_MNLR\_v3\_DW.TransportDelay1\_IWORK.Head == Code MNLR v3 DW.TransportDelay1 IWORK.Tail) { Code\_MNLR\_v3\_DW.TransportDelay1\_IWORK.Tail = ((Code\_MNLR\_v3\_DW.TransportDelay1\_IWORK.Tail < (Code MNLR v3 DW.TransportDelay1 IWORK.CircularBufSize-1))? (Code\_MNLR\_v3\_DW.TransportDelay1\_IWORK.Tail+1): 0); (\*uBuffer + Code MNLR v3 DW.TransportDelav1 IWORK.CircularBufSize) [Code\_MNLR\_v3\_DW.TransportDelay1\_IWORK.Head] = simTime; (\*uBuffer)[Code MNLR v3 DW.TransportDelay1 IWORK.Head] = Code\_MNLR\_v3\_B.OUTPUT\_1\_0[27]; } /\* Update for TransportDelay: '<Root>/Transport Delay16' \*/ real\_T \*\*uBuffer = (real\_T\*\*) { &Code MNLR v3 DW.TransportDelay16 PWORK.TUbufferPtrs[0]; real T simTime = Code MNLR v3 M->Timing.t[0]; Code\_MNLR\_v3\_DW.TransportDelay16\_IWORK.Head = ((Code MNLR v3 DW.TransportDelay16 IWORK.Head < (Code\_MNLR\_v3\_DW.TransportDelay16\_IWORK.CircularBufSize-1)) ? (Code\_MNLR\_v3\_DW.TransportDelay16\_IWORK.Head+1): 0); if (Code MNLR v3 DW.TransportDelay16 IWORK.Head == Code\_MNLR\_v3\_DW.TransportDelay16\_IWORK.Tail) { Code MNLR v3 DW.TransportDelay16 IWORK.Tail = ((Code MNLR v3 DW.TransportDelav16 IWORK.Tail < (Code\_MNLR\_v3\_DW.TransportDelay16\_IWORK.CircularBufSize-1)) ? (Code MNLR v3 DW.TransportDelay16 IWORK.Tail+1): 0); (\*uBuffer + Code\_MNLR\_v3\_DW.TransportDelay16\_IWORK.CircularBufSize) [Code MNLR v3 DW.TransportDelay16 IWORK.Head] = simTime; (\*uBuffer)[Code\_MNLR\_v3\_DW.TransportDelay16\_IWORK.Head] = Code MNLR v3 B.OUTPUT 1 0[20]; /\* Update for TransportDelay: '<Root>/Transport Delay2' \*/ { real T \*\*uBuffer = (real T\*\*)&Code MNLR v3 DW.TransportDelay2 PWORK.TUbufferPtrs[0]; real\_T simTime = Code\_MNLR\_v3\_M->Timing.t[0]; Code\_MNLR\_v3\_DW.TransportDelay2\_IWORK.Head = ((Code MNLR v3 DW.TransportDelay2 IWORK.Head < (Code MNLR v3 DW.TransportDelay2 IWORK.CircularBufSize-1))? (Code\_MNLR\_v3\_DW.TransportDelay2\_IWORK.Head+1): 0); if (Code MNLR v3 DW.TransportDelay2 IWORK.Head == Code\_MNLR\_v3\_DW.TransportDelay2\_IWORK.Tail) { Code MNLR v3 DW.TransportDelay2 IWORK.Tail = ((Code MNLR v3 DW.TransportDelay2 IWORK.Tail < (Code\_MNLR\_v3\_DW.TransportDelay2\_IWORK.CircularBufSize-1)) ? (Code MNLR v3 DW.TransportDelay2 IWORK.Tail+1):0); (\*uBuffer + Code MNLR v3 DW.TransportDelay2 IWORK.CircularBufSize) [Code MNLR v3 DW.TransportDelay2 IWORK.Head] = simTime; (\*uBuffer)[Code\_MNLR\_v3\_DW.TransportDelay2\_IWORK.Head] = Code\_MNLR\_v3\_B.OUTPUT\_1\_0[23];

}

/\* Update for TransportDelay: '<Root>/Transport Delay3' \*/ real\_T \*\*uBuffer = (real\_T\*\*) &Code\_MNLR\_v3\_DW.TransportDelay3\_PWORK.TUbufferPtrs[0]; real T simTime = Code MNLR v3 M->Timing.t[0]; Code\_MNLR\_v3\_DW.TransportDelay3\_IWORK.Head = ((Code MNLR v3 DW.TransportDelay3 IWORK.Head < (Code\_MNLR\_v3\_DW.TransportDelay3\_IWORK.CircularBufSize-1)) ? (Code\_MNLR\_v3\_DW.TransportDelay3\_IWORK.Head+1): 0); if (Code MNLR v3 DW.TransportDelay3 IWORK.Head == Code\_MNLR\_v3\_DW.TransportDelay3\_IWORK.Tail) { Code\_MNLR\_v3\_DW.TransportDelay3\_IWORK.Tail = ((Code MNLR v3 DW.TransportDelay3 IWORK.Tail < (Code\_MNLR\_v3\_DW.TransportDelay3\_IWORK.CircularBufSize-1)) ? (Code MNLR v3 DW.TransportDelay3 IWORK.Tail+1):0); (\*uBuffer + Code MNLR v3 DW.TransportDelay3 IWORK.CircularBufSize) [Code MNLR v3 DW.TransportDelay3 IWORK.Head] = simTime; (\*uBuffer)[Code\_MNLR\_v3\_DW.TransportDelay3\_IWORK.Head] = Code MNLR v3 B.OUTPUT 1 0[24]; /\* Update for TransportDelay: '<Root>/Transport Delay4' \*/ { real T \*\*uBuffer = (real T\*\*)&Code\_MNLR\_v3\_DW.TransportDelay4\_PWORK.TUbufferPtrs[0]; real T simTime = Code MNLR v3 M->Timing.t[0]; Code\_MNLR\_v3\_DW.TransportDelay4\_IWORK.Head = ((Code MNLR v3 DW.TransportDelay4 IWORK.Head < (Code MNLR v3 DW.TransportDelay4 IWORK.CircularBufSize-1))? (Code\_MNLR\_v3\_DW.TransportDelay4\_IWORK.Head+1): 0); if (Code MNLR v3 DW.TransportDelay4 IWORK.Head == Code\_MNLR\_v3\_DW.TransportDelay4\_IWORK.Tail) Code MNLR v3 DW.TransportDelay4 IWORK.Tail = ((Code MNLR v3 DW.TransportDelay4 IWORK.Tail < (Code\_MNLR\_v3\_DW.TransportDelay4\_IWORK.CircularBufSize-1)) ? (Code MNLR v3 DW.TransportDelay4 IWORK.Tail+1):0); (\*uBuffer + Code MNLR v3 DW.TransportDelay4 IWORK.CircularBufSize) [Code MNLR v3 DW.TransportDelay4 IWORK.Head] = simTime; (\*uBuffer)[Code\_MNLR\_v3\_DW.TransportDelay4\_IWORK.Head] = Code\_MNLR\_v3\_B.OUTPUT\_1\_0[25]; /\* Update for TransportDelay: '<Root>/Transport Delay5' \*/ ł real T \*\*uBuffer = (real T\*\*)&Code\_MNLR\_v3\_DW.TransportDelay5\_PWORK.TUbufferPtrs[0]; real T simTime = Code MNLR v3 M->Timing.t[0]; Code\_MNLR\_v3\_DW.TransportDelay5\_IWORK.Head = ((Code MNLR v3 DW.TransportDelay5 IWORK.Head < (Code\_MNLR\_v3\_DW.TransportDelay5\_IWORK.CircularBufSize-1)) ? (Code\_MNLR\_v3\_DW.TransportDelay5\_IWORK.Head+1): 0); if (Code MNLR v3 DW.TransportDelay5 IWORK.Head == Code\_MNLR\_v3\_DW.TransportDelay5\_IWORK.Tail) { Code MNLR v3 DW.TransportDelay5 IWORK.Tail = ((Code MNLR v3 DW.TransportDelay5 IWORK.Tail < (Code\_MNLR\_v3\_DW.TransportDelay5\_IWORK.CircularBufSize-1)) ? (Code\_MNLR\_v3\_DW.TransportDelay5\_IWORK.Tail+1): 0); }

(\*uBuffer + Code\_MNLR\_v3\_DW.TransportDelay5\_IWORK.CircularBufSize) [Code MNLR v3 DW.TransportDelay5 IWORK.Head] = simTime; (\*uBuffer)[Code\_MNLR\_v3\_DW.TransportDelay5\_IWORK.Head] = Code\_MNLR\_v3\_B.OUTPUT\_1\_0[30]; /\* Update for TransportDelay: '<Root>/Transport Delay6' \*/ real T \*\*uBuffer = (real T\*\*)&Code\_MNLR\_v3\_DW.TransportDelay6\_PWORK.TUbufferPtrs[0]; real T simTime = Code MNLR v3 M->Timing.t[0]; Code\_MNLR\_v3\_DW.TransportDelay6\_IWORK.Head = ((Code MNLR v3 DW.TransportDelay6 IWORK.Head < (Code MNLR v3 DW.TransportDelay6 IWORK.CircularBufSize-1)) ? (Code\_MNLR\_v3\_DW.TransportDelay6\_IWORK.Head+1): 0); if (Code MNLR v3 DW.TransportDelay6 IWORK.Head == Code\_MNLR\_v3\_DW.TransportDelay6\_IWORK.Tail) { Code MNLR v3 DW.TransportDelay6 IWORK.Tail = ((Code MNLR v3 DW.TransportDelay6 IWORK.Tail < (Code\_MNLR\_v3\_DW.TransportDelay6\_IWORK.CircularBufSize-1)) ? (Code MNLR v3 DW.TransportDelay6 IWORK.Tail+1):0); (\*uBuffer + Code\_MNLR\_v3\_DW.TransportDelay6\_IWORK.CircularBufSize) [Code MNLR v3 DW.TransportDelay6 IWORK.Head] = simTime; (\*uBuffer)[Code\_MNLR\_v3\_DW.TransportDelay6\_IWORK.Head] = Code\_MNLR\_v3\_B.OUTPUT\_1\_0[26]; /\* Update for TransportDelay: '<Root>/Transport Delay7' \*/ real T \*\*uBuffer = (real T\*\*)&Code\_MNLR\_v3\_DW.TransportDelay7\_PWORK.TUbufferPtrs[0]; real T simTime = Code MNLR v3 M->Timing.t[0]; Code\_MNLR\_v3\_DW.TransportDelay7\_IWORK.Head = ((Code\_MNLR\_v3\_DW.TransportDelay7\_IWORK.Head < (Code MNLR v3 DW.TransportDelay7 IWORK.CircularBufSize-1))? (Code\_MNLR\_v3\_DW.TransportDelay7\_IWORK.Head+1): 0); if (Code MNLR v3 DW.TransportDelay7 IWORK.Head == Code\_MNLR\_v3\_DW.TransportDelay7\_IWORK.Tail) { Code MNLR v3 DW.TransportDelay7 IWORK.Tail = ((Code MNLR v3 DW.TransportDelay7 IWORK.Tail < (Code\_MNLR\_v3\_DW.TransportDelay7\_IWORK.CircularBufSize-1)) ? (Code\_MNLR\_v3\_DW.TransportDelay7\_IWORK.Tail+1): 0); (\*uBuffer + Code\_MNLR\_v3\_DW.TransportDelay7\_IWORK.CircularBufSize) [Code MNLR v3 DW.TransportDelay7 IWORK.Head] = simTime; (\*uBuffer)[Code MNLR v3 DW.TransportDelay7 IWORK.Head] = Code\_MNLR\_v3\_B.OUTPUT\_1\_0[28]; /\* Update for TransportDelay: '<Root>/Transport Delay8' \*/ ł real T \*\*uBuffer = (real T\*\*)&Code\_MNLR\_v3\_DW.TransportDelay8\_PWORK.TUbufferPtrs[0]; real T simTime = Code MNLR v3 M->Timing.t[0]; Code MNLR v3 DW.TransportDelay8 IWORK.Head = ((Code MNLR v3 DW.TransportDelay8 IWORK.Head < (Code MNLR v3 DW.TransportDelay8 IWORK.CircularBufSize-1))? (Code\_MNLR\_v3\_DW.TransportDelay8\_IWORK.Head+1): 0); if (Code\_MNLR\_v3\_DW.TransportDelay8\_IWORK.Head == Code\_MNLR\_v3\_DW.TransportDelay8\_IWORK.Tail) {

Code\_MNLR\_v3\_DW.TransportDelay8\_IWORK.Tail = ((Code MNLR v3 DW.TransportDelay8 IWORK.Tail < (Code\_MNLR\_v3\_DW.TransportDelay8\_IWORK.CircularBufSize-1)) ? (Code\_MNLR\_v3\_DW.TransportDelay8\_IWORK.Tail+1): 0); (\*uBuffer + Code\_MNLR\_v3\_DW.TransportDelay8\_IWORK.CircularBufSize) [Code MNLR v3 DW.TransportDelay8 IWORK.Head] = simTime; (\*uBuffer)[Code\_MNLR\_v3\_DW.TransportDelay8\_IWORK.Head] = Code\_MNLR\_v3\_B.OUTPUT\_1\_0[29]; } /\* Update for SimscapeExecutionBlock: '<S954>/STATE 1' \*/ simulationData = (NeslSimulationData \*)Code MNLR v3 DW.STATE 1 SimData; time = Code\_MNLR\_v3\_M->Timing.t[0]; simulationData->mData->mTime.mN = 1; simulationData->mData->mTime.mX = &time; simulationData->mData->mContStates.mN = 245; simulationData->mData->mContStates.mX = &Code\_MNLR\_v3\_X.Code\_MNLR\_v3Double\_Acting\_Hydra[0]; simulationData->mDiscStates.mN = 0; simulationData->mData->mDiscStates.mX = &Code MNLR v3 DW.STATE 1 Discrete; simulationData->mData->mModeVector.mN = 194; simulationData->mData->mModeVector.mX = &Code MNLR v3 DW.STATE 1 Modes[0]; tmp = false;simulationData->mData->mFoundZcEvents = tmp; simulationData->mData->mIsMajorTimeStep = rtmIsMajorTimeStep(Code MNLR v3 M); tmp = false;simulationData->mData->mIsSolverAssertCheck = tmp; simulationData->mData->mIsSolverCheckingCIC = false; tmp = rtsiIsSolverComputingJacobian(&Code\_MNLR\_v3\_M->solverInfo); simulationData->mData->mIsComputingJacobian = tmp; simulationData->mData->mIsEvaluatingF0 = false; simulationData->mData->mIsSolverRequestingReset = false; simulationData->mData->mIsModeUpdateTimeStep = rtsiIsModeUpdateTimeStep (&Code\_MNLR\_v3\_M->solverInfo); tmp 1[0] = 0; $tmp_0[0] = Code_MNLR_v3_B.INPUT_12_1_1[0];$ tmp\_0[1] = Code\_MNLR\_v3\_B.INPUT\_12\_1\_1[1]; tmp 0[2] = Code MNLR v3 B.INPUT 12 1 1[2]; $tmp_0[3] = Code_MNLR_v3_B.INPUT_12_1_1[3];$  $tmp_1[1] = 4;$  $tmp_0[4] = Code_MNLR_v3_B.INPUT_1_1_1[0];$ tmp\_0[5] = Code\_MNLR\_v3\_B.INPUT\_1\_1\_1[1]; tmp 0[6] = Code MNLR v3 B.INPUT 1 1 1[2]; $tmp_0[7] = Code_MNLR_v3_B.INPUT_1_1_1[3];$  $tmp_1[2] = 8;$ tmp 0[8] = Code MNLR v3 B.INPUT 13 1 1[0]; $tmp_0[9] = Code_MNLR_v3_B.INPUT_13_1_1[1];$  $tmp_0[10] = Code_MNLR_v3_B.INPUT_13_1_1[2];$  $tmp_0[11] = Code_MNLR_v3_B.INPUT_13_1_1[3];$  $tmp_1[3] = 12;$ tmp 0[12] = Code MNLR v3 B.INPUT 15 1 1[0]; $tmp_0[13] = Code_MNLR_v3_B.INPUT_15_1_1[1];$  $tmp_0[14] = Code_MNLR_v3_B.INPUT_15_1_1[2];$  $tmp_0[15] = Code_MNLR_v3_B.INPUT_15_1_1[3];$  $tmp_1[4] = 16;$  $tmp_0[16] = Code_MNLR_v3_B.INPUT_21_1_1[0];$  $tmp_0[17] = Code_MNLR_v3_B.INPUT_21_1_1[1];$ 

 $tmp_0[18] = Code_MNLR_v3_B.INPUT_21_1_1[2];$  $tmp_0[19] = Code_MNLR_v3_B.INPUT_21_1_1[3];$  $tmp_1[5] = 20;$  $tmp_0[20] = Code_MNLR_v3_B.INPUT_24_1_1[0];$ tmp\_0[21] = Code\_MNLR\_v3\_B.INPUT\_24\_1\_1[1];  $tmp_0[22] = Code_MNLR_v3_B.INPUT_24_1_1[2];$  $tmp_0[23] = Code_MNLR_v3_B.INPUT_24_1_1[3];$  $tmp_1[6] = 24;$  $tmp_0[24] = Code_MNLR_v3_B.INPUT_18_1_1[0];$ tmp\_0[25] = Code\_MNLR\_v3\_B.INPUT\_18\_1\_1[1];  $tmp_0[26] = Code_MNLR_v3_B.INPUT_18_1_1[2];$ tmp 0[27] = Code MNLR v3 B.INPUT 18 1 1[3];tmp 1[7] = 28; $tmp_0[28] = Code_MNLR_v3_B.INPUT_26_1_1[0];$ tmp\_0[29] = Code\_MNLR\_v3\_B.INPUT\_26\_1\_1[1];  $tmp_0[30] = Code_MNLR_v3_B.INPUT_26_1_1[2];$  $tmp_0[31] = Code_MNLR_v3_B.INPUT_26_1_1[3];$  $tmp_1[8] = 32;$  $tmp_0[32] = Code_MNLR_v3_B.INPUT_29_1_1[0];$ tmp\_0[33] = Code\_MNLR\_v3\_B.INPUT\_29\_1\_1[1]; tmp\_0[34] = Code\_MNLR\_v3\_B.INPUT\_29\_1\_1[2]; tmp\_0[35] = Code\_MNLR\_v3\_B.INPUT\_29\_1\_1[3]; tmp 1[9] = 36; $tmp_0[36] = Code_MNLR_v3_B.INPUT_31_1_1[0];$  $tmp_0[37] = Code_MNLR_v3_B.INPUT_31_1_1[1];$ tmp\_0[38] = Code\_MNLR\_v3\_B.INPUT\_31\_1\_1[2]; tmp\_0[39] = Code\_MNLR\_v3\_B.INPUT\_31\_1\_1[3]; tmp 1[10] = 40; $tmp_0[40] = Code_MNLR_v3_B.INPUT_2_1_1[0];$  $tmp_0[41] = Code_MNLR_v3_B.INPUT_2_1_1[1];$  $tmp_0[42] = Code_MNLR_v3_B.INPUT_2_1_1[2];$  $tmp_0[43] = Code_MNLR_v3_B.INPUT_2_1_1[3];$ tmp 1[11] = 44;tmp 0[44] = Code MNLR v3 B.INPUT 28 1 1[0];tmp\_0[45] = Code\_MNLR\_v3\_B.INPUT\_28\_1\_1[1]; tmp 0[46] = Code MNLR v3 B.INPUT 28 1 1[2]; $tmp_0[47] = Code_MNLR_v3_B.INPUT_28_1_1[3];$ tmp 1[12] = 48;tmp 0[48] = Code MNLR v3 B.INPUT 5 1 1[0]; $tmp_0[49] = Code_MNLR_v3_B.INPUT_5_1_1[1];$ tmp\_0[50] = Code\_MNLR\_v3\_B.INPUT\_5\_1\_1[2];  $tmp_0[51] = Code_MNLR_v3_B.INPUT_5_1_1[3];$  $tmp_1[13] = 52;$ tmp 0[52] = Code MNLR v3 B.INPUT 7 1 1[0]; $tmp_0[53] = Code_MNLR_v3_B.INPUT_7_1_1[1];$  $tmp_0[54] = Code_MNLR_v3_B.INPUT_7_1_1[2];$  $tmp_0[55] = Code_MNLR_v3_B.INPUT_7_1_1[3];$  $tmp_1[14] = 56;$  $tmp_0[56] = Code_MNLR_v3_B.INPUT_9_1_1[0];$  $tmp_0[57] = Code_MNLR_v3_B.INPUT_9_1_1[1];$ tmp\_0[58] = Code\_MNLR\_v3\_B.INPUT\_9\_1\_1[2]; tmp 0[59] = Code MNLR v3 B.INPUT 9 1 1[3];tmp 1[15] = 60;  $tmp_0[60] = Code_MNLR_v3_B.INPUT_17_1_1[0];$ tmp 0[61] = Code MNLR v3 B.INPUT 17 1 1[1]; $tmp_0[62] = Code_MNLR_v3_B.INPUT_17_1_1[2];$  $tmp_0[63] = Code_MNLR_v3_B.INPUT_17_1_1[3];$  $tmp_1[16] = 64;$ 

 $tmp_0[64] = Code_MNLR_v3_B.INPUT_3_1_1[0];$  $tmp_0[65] = Code_MNLR_v3_B.INPUT_3_1_1[1];$  $tmp_0[66] = Code_MNLR_v3_B.INPUT_3_1_1[2];$  $tmp_0[67] = Code_MNLR_v3_B.INPUT_3_1_1[3];$  $tmp_1[17] = 68;$  $tmp_0[68] = Code_MNLR_v3_B.INPUT_4_1_1[0];$ tmp 0[69] = Code MNLR v3 B.INPUT 4 1 1[1]; $tmp_0[70] = Code_MNLR_v3_B.INPUT_4_1_1[2];$  $tmp_0[71] = Code_MNLR_v3_B.INPUT_4_1_1[3];$ tmp 1[18] = 72; $tmp_0[72] = Code_MNLR_v3_B.INPUT_6_1_1[0];$ tmp 0[73] = Code MNLR v3 B.INPUT 6 1 1[1];tmp\_0[74] = Code\_MNLR\_v3\_B.INPUT\_6\_1\_1[2];  $tmp_0[75] = Code_MNLR_v3_B.INPUT_6_1_1[3];$ tmp 1[19] = 76; $tmp_0[76] = Code_MNLR_v3_B.INPUT_8_1_1[0];$  $tmp_0[77] = Code_MNLR_v3_B.INPUT_8_1_1[1];$  $tmp_0[78] = Code_MNLR_v3_B.INPUT_8_1_1[2];$  $tmp_0[79] = Code_MNLR_v3_B.INPUT_8_1_1[3];$ tmp 1[20] = 80:  $tmp_0[80] = Code_MNLR_v3_B.INPUT_10_1_1[0];$  $tmp_0[81] = Code_MNLR_v3_B.INPUT_10_1_1[1];$ tmp 0[82] = Code MNLR v3 B.INPUT 10 1 1[2]; $tmp_0[83] = Code_MNLR_v3_B.INPUT_10_1_1[3];$  $tmp_1[21] = 84;$  $tmp_0[84] = Code_MNLR_v3_B.INPUT_11_1_1[0];$ tmp\_0[85] = Code\_MNLR\_v3\_B.INPUT\_11\_1[1]; tmp 0[86] = Code MNLR v3 B.INPUT 11 1 1[2]; $tmp_0[87] = Code_MNLR_v3_B.INPUT_11_1[3];$  $tmp_1[22] = 88;$ tmp 0[88] = Code MNLR v3 B.INPUT 14 1 1[0]; $tmp_0[89] = Code_MNLR_v3_B.INPUT_14_1_1[1];$  $tmp_0[90] = Code_MNLR_v3_B.INPUT_14_1_1[2];$  $tmp_0[91] = Code_MNLR_v3_B.INPUT_14_1_1[3];$  $tmp_1[23] = 92;$ tmp 0[92] = Code MNLR v3 B.INPUT 16 1 1[0]; $tmp_0[93] = Code_MNLR_v3_B.INPUT_16_1_1[1];$  $tmp_0[94] = Code_MNLR_v3_B.INPUT_16_1_1[2];$  $tmp_0[95] = Code_MNLR_v3_B.INPUT_16_1_1[3];$  $tmp_1[24] = 96;$ tmp\_0[96] = Code\_MNLR\_v3\_B.INPUT\_19\_1\_1[0]; tmp\_0[97] = Code\_MNLR\_v3\_B.INPUT\_19\_1\_1[1]; tmp\_0[98] = Code\_MNLR\_v3\_B.INPUT\_19\_1\_1[2]; tmp 0[99] = Code MNLR v3 B.INPUT 19 1 1[3];tmp 1[25] = 100; $tmp_0[100] = Code_MNLR_v3_B.INPUT_20_1_1[0];$  $tmp_0[101] = Code_MNLR_v3_B.INPUT_20_1_1[1];$  $tmp_0[102] = Code_MNLR_v3_B.INPUT_20_1_1[2];$  $tmp_0[103] = Code_MNLR_v3_B.INPUT_20_1_1[3];$  $tmp_1[26] = 104;$ tmp\_0[104] = Code\_MNLR\_v3\_B.INPUT\_22\_1\_1[0]; tmp 0[105] = Code MNLR v3 B.INPUT 22 1 1[1];  $tmp_0[106] = Code_MNLR_v3_B.INPUT_22_1_1[2];$  $tmp_0[107] = Code_MNLR_v3_B.INPUT_22_1_1[3];$ tmp 1[27] = 108; $tmp_0[108] = Code_MNLR_v3_B.INPUT_25_1_1[0];$ tmp\_0[109] = Code\_MNLR\_v3\_B.INPUT\_25\_1\_1[1];  $tmp_0[110] = Code_MNLR_v3_B.INPUT_25_1_1[2];$ 

```
tmp_0[111] = Code_MNLR_v3_B.INPUT_25_1_1[3];
 tmp_1[28] = 112;
 tmp_0[112] = Code_MNLR_v3_B.INPUT_23_1_1[0];
 tmp_0[113] = Code_MNLR_v3_B.INPUT_23_1_1[1];
 tmp_0[114] = Code_MNLR_v3_B.INPUT_23_1_1[2];
 tmp_0[115] = Code_MNLR_v3_B.INPUT_23_1_1[3];
 tmp 1[29] = 116;
 tmp_0[116] = Code_MNLR_v3_B.INPUT_27_1_1[0];
 tmp_0[117] = Code_MNLR_v3_B.INPUT_27_1_1[1];
 tmp_0[118] = Code_MNLR_v3_B.INPUT_27_1_1[2];
 tmp_0[119] = Code_MNLR_v3_B.INPUT_27_1_1[3];
 tmp 1[30] = 120;
 tmp 0[120] = Code MNLR v3 B.INPUT 30 1 1[0];
 tmp_0[121] = Code_MNLR_v3_B.INPUT_30_1_1[1];
 tmp_0[122] = Code_MNLR_v3_B.INPUT_30_1_1[2];
 tmp_0[123] = Code_MNLR_v3_B.INPUT_30_1_1[3];
 tmp 1[31] = 124;
 tmp 0[124] = Code MNLR v3 B.INPUT 32 1 1[0];
 tmp_0[125] = Code_MNLR_v3_B.INPUT_32_1_1[1];
 tmp 0[126] = Code MNLR v3 B.INPUT 32 1 1[2];
 tmp 0[127] = Code MNLR v3 B.INPUT 32 1 1[3];
 tmp_1[32] = 128;
 tmp 0[128] = Code MNLR v3 B.INPUT 33 1 1[0];
 tmp_0[129] = Code_MNLR_v3_B.INPUT_33_1_1[1];
 tmp_0[130] = Code_MNLR_v3_B.INPUT_33_1_1[2];
 tmp_0[131] = Code_MNLR_v3_B.INPUT_33_1_1[3];
 tmp_1[33] = 132;
 simulationData->mData->mInputValues.mN = 132;
 simulationData->mData->mInputValues.mX = &tmp 0[0];
 simulationData->mData->mInputOffsets.mN = 34;
 simulationData->mData->mInputOffsets.mX = &tmp 1[0];
 diagnosticManager = (NeuDiagnosticManager *)Code_MNLR_v3_DW.STATE_1_DiagMgr;
 diagnosticTree = neu_diagnostic_manager_get_initial_tree(diagnosticManager);
 tmp 2 = ne simulator method((NeslSimulator *))
  Code_MNLR_v3_DW.STATE_1_Simulator, NESL_SIM_UPDATE, simulationData,
  diagnosticManager);
 if (tmp_2 != 0) {
  tmp = error buffer is empty(rtmGetErrorStatus(Code MNLR v3 M));
  if (tmp) {
   msg = rtw_diagnostics_msg(diagnosticTree);
   rtmSetErrorStatus(Code_MNLR_v3_M, msg);
  ł
 ł
 /* End of Update for SimscapeExecutionBlock: '<S954>/STATE 1' */
                    /* end MajorTimeStep */
if (rtmIsMajorTimeStep(Code_MNLR_v3_M)) {
 rt ertODEUpdateContinuousStates(&Code MNLR v3 M->solverInfo);
 /* Update absolute time for base rate */
 /* The "clockTick0" counts the number of times the code of this task has
 * been executed. The absolute time is the multiplication of "clockTick0"
 * and "Timing.stepSize0". Size of "clockTick0" ensures timer will not
 * overflow during the application lifespan selected.
 ++Code_MNLR_v3_M->Timing.clockTick0;
 Code MNLR v3 M->Timing.t[0] = rtsiGetSolverStopTime
  (&Code_MNLR_v3_M->solverInfo);
  /* Update absolute timer for sample time: [0.17482517482517482s, 0.0s] */
```

/\* The "clockTick1" counts the number of times the code of this task has \* been executed. The resolution of this integer timer is 0.17482517482517482, which is the step \* of the task. Size of "clockTick1" ensures timer will not overflow during the \* application lifespan selected. \*/ Code\_MNLR\_v3\_M->Timing.clockTick1++; } /\* end MajorTimeStep \*/ } /\* Derivatives for root system: '<Root>' \*/ void Code MNLR v3 derivatives(void) NeslSimulationData \*simulationData; NeuDiagnosticManager \*diagnosticManager; NeuDiagnosticTree \*diagnosticTree; XDot\_Code\_MNLR\_v3\_T \*\_rtXdot; char \*msg; real\_T tmp\_0[132]; real T time; int32\_T tmp\_2; int\_T tmp\_1[34]; boolean T tmp; rtXdot = ((XDot\_Code\_MNLR\_v3\_T \*) Code\_MNLR\_v3\_M->derivs); /\* Derivatives for Integrator: '<S214>/Integrator' \*/ \_rtXdot->Integrator\_CSTATE = Code\_MNLR\_v3\_B.IntegralGain\_en; /\* Derivatives for Integrator: '<S209>/Filter' \*/ rtXdot->Filter CSTATE = Code MNLR v3 B.FilterCoefficient; /\* Derivatives for Integrator: '<S886>/Integrator' \*/ \_rtXdot->Integrator\_CSTATE\_p = Code\_MNLR\_v3\_B.IntegralGain\_k; /\* Derivatives for Integrator: '<S881>/Filter' \*/ \_rtXdot->Filter\_CSTATE\_p = Code\_MNLR\_v3\_B.FilterCoefficient\_n; /\* Derivatives for Integrator: '<S262>/Integrator' \*/ \_rtXdot->Integrator\_CSTATE\_n = Code\_MNLR\_v3\_B.IntegralGain\_b; /\* Derivatives for Integrator: '<S257>/Filter' \*/ \_rtXdot->Filter\_CSTATE\_i = Code\_MNLR\_v3\_B.FilterCoefficient 1; /\* Derivatives for Integrator: '<S310>/Integrator' \*/ \_rtXdot->Integrator\_CSTATE\_j = Code\_MNLR\_v3\_B.IntegralGain\_c; /\* Derivatives for Integrator: '<S305>/Filter' \*/ \_rtXdot->Filter\_CSTATE\_k = Code\_MNLR\_v3\_B.FilterCoefficient\_m; /\* Derivatives for Integrator: '<S358>/Integrator' \*/ \_rtXdot->Integrator\_CSTATE\_m = Code\_MNLR\_v3\_B.IntegralGain\_o3; /\* Derivatives for Integrator: '<S353>/Filter' \*/ rtXdot->Filter CSTATE h = Code MNLR v3 B.FilterCoefficient a; /\* Derivatives for Integrator: '<S502>/Integrator' \*/ \_rtXdot->Integrator\_CSTATE\_l = Code\_MNLR\_v3\_B.IntegralGain\_f; /\* Derivatives for Integrator: '<S497>/Filter' \*/ \_rtXdot->Filter\_CSTATE\_a = Code\_MNLR\_v3\_B.FilterCoefficient\_p; /\* Derivatives for Integrator: '<\$406>/Integrator' \*/ \_rtXdot->Integrator\_CSTATE\_ma = Code\_MNLR\_v3\_B.IntegralGain\_em; /\* Derivatives for Integrator: '<S401>/Filter' \*/ \_rtXdot->Filter\_CSTATE\_g = Code\_MNLR\_v3\_B.FilterCoefficient\_i; /\* Derivatives for Integrator: '<S550>/Integrator' \*/ rtXdot->Integrator CSTATE is = Code MNLR v3 B.IntegralGain a; /\* Derivatives for Integrator: '<S545>/Filter' \*/ \_rtXdot->Filter\_CSTATE\_gy = Code\_MNLR\_v3\_B.FilterCoefficient\_ab;

size

\_rtXdot->Integrator\_CSTATE\_a = Code\_MNLR\_v3\_B.IntegralGain\_g; /\* Derivatives for Integrator: '<S449>/Filter' \*/ \_rtXdot->Filter\_CSTATE\_ay = Code\_MNLR\_v3\_B.FilterCoefficient\_e; /\* Derivatives for Integrator: '<S166>/Integrator' \*/ \_rtXdot->Integrator\_CSTATE\_f = Code\_MNLR\_v3\_B.IntegralGain\_i; /\* Derivatives for Integrator: '<S161>/Filter' \*/ rtXdot->Filter CSTATE n = Code MNLR v3 B.FilterCoefficient ni; /\* Derivatives for Integrator: '<S598>/Integrator' \*/ \_rtXdot->Integrator\_CSTATE\_i = Code\_MNLR\_v3\_B.IntegralGain\_m; /\* Derivatives for Integrator: '<S593>/Filter' \*/ \_rtXdot->Filter\_CSTATE\_o = Code\_MNLR\_v3\_B.FilterCoefficient\_k; /\* Derivatives for Integrator: '<S646>/Integrator' \*/ rtXdot->Integrator CSTATE k = Code MNLR v3 B.IntegralGain e;/\* Derivatives for Integrator: '<S641>/Filter' \*/ \_rtXdot->Filter\_CSTATE\_pb = Code\_MNLR\_v3\_B.FilterCoefficient\_me; /\* Derivatives for TransferFcn: '<S103>/Transfer Fcn7' \*/ \_rtXdot->TransferFcn7\_CSTATE = 0.0; rtXdot->TransferFcn7 CSTATE += -0.0 \* Code MNLR v3 X.TransferFcn7 CSTATE; \_rtXdot->TransferFcn7\_CSTATE += Code\_MNLR\_v3\_B.Gain6; /\* Derivatives for Integrator: '<\$742>/Integrator' \*/ \_rtXdot->Integrator\_CSTATE\_jt = Code\_MNLR\_v3\_B.IntegralGain\_o; /\* Derivatives for Integrator: '<S737>/Filter' \*/ rtXdot->Filter CSTATE d = Code MNLR v3 B.FilterCoefficient i2; /\* Derivatives for Integrator: '<S694>/Integrator' \*/ \_rtXdot->Integrator\_CSTATE\_kg = Code\_MNLR\_v3\_B.IntegralGain\_d; /\* Derivatives for Integrator: '<S689>/Filter' \*/ \_rtXdot->Filter\_CSTATE\_i4 = Code\_MNLR\_v3\_B.FilterCoefficient\_c; /\* Derivatives for Integrator: '<S790>/Integrator' \*/ \_rtXdot->Integrator\_CSTATE\_j2 = Code\_MNLR\_v3\_B.IntegralGain\_l; /\* Derivatives for Integrator: '<S785>/Filter' \*/ \_rtXdot->Filter\_CSTATE\_j = Code\_MNLR\_v3\_B.FilterCoefficient\_c0; /\* Derivatives for Integrator: '<S838>/Integrator' \*/ \_rtXdot->Integrator\_CSTATE\_ft = Code\_MNLR\_v3\_B.IntegralGain; /\* Derivatives for Integrator: '<S833>/Filter' \*/ \_rtXdot->Filter\_CSTATE\_ku = Code\_MNLR\_v3\_B.FilterCoefficient\_al; /\* Derivatives for SimscapeExecutionBlock: '<S954>/STATE 1' \*/ simulationData = (NeslSimulationData \*)Code MNLR v3 DW.STATE 1 SimData; time = Code MNLR v3 M->Timing.t[0]; simulationData->mData->mTime.mN = 1; simulationData->mData->mTime.mX = &time; simulationData->mData->mContStates.mN = 245; simulationData->mData->mContStates.mX = &Code\_MNLR\_v3\_X.Code\_MNLR\_v3Double\_Acting\_Hydra[0]; simulationData->mDiscStates.mN = 0; simulationData->mData->mDiscStates.mX = &Code\_MNLR\_v3\_DW.STATE\_1\_Discrete; simulationData->mData->mModeVector.mN = 194; simulationData->mData->mModeVector.mX = &Code MNLR v3 DW.STATE 1 Modes[0]; tmp = false;simulationData->mData->mFoundZcEvents = tmp; simulationData->mData->mIsMajorTimeStep = rtmIsMajorTimeStep(Code\_MNLR\_v3\_M); tmp = false;simulationData->mIsSolverAssertCheck = tmp; simulationData->mData->mIsSolverCheckingCIC = false; tmp = rtsiIsSolverComputingJacobian(&Code MNLR v3 M->solverInfo); simulationData->mData->mIsComputingJacobian = tmp; simulationData->mData->mIsEvaluatingF0 = false; simulationData->mData->mIsSolverRequestingReset = false; simulationData->mData->mIsModeUpdateTimeStep = rtsiIsModeUpdateTimeStep

tmp 1[0] = 0; $tmp_0[0] = Code_MNLR_v3_B.INPUT_12_1_1[0];$  $tmp_0[1] = Code_MNLR_v3_B.INPUT_12_1_1[1];$  $tmp_0[2] = Code_MNLR_v3_B.INPUT_12_1_1[2];$  $tmp_0[3] = Code_MNLR_v3_B.INPUT_12_1_1[3];$ tmp 1[1] = 4; $tmp_0[4] = Code_MNLR_v3_B.INPUT_1_1_1[0];$ tmp\_0[5] = Code\_MNLR\_v3\_B.INPUT\_1\_1\_1[1];  $tmp_0[6] = Code_MNLR_v3_B.INPUT_1_1_1[2];$  $tmp_0[7] = Code_MNLR_v3_B.INPUT_1_1_1[3];$ tmp 1[2] = 8;tmp 0[8] = Code MNLR v3 B.INPUT 13 1 1[0]; $tmp_0[9] = Code_MNLR_v3_B.INPUT_13_1_1[1];$ tmp 0[10] = Code MNLR v3 B.INPUT 13 1 1[2]; $tmp_0[11] = Code_MNLR_v3_B.INPUT_13_1_1[3];$  $tmp_1[3] = 12;$  $tmp_0[12] = Code_MNLR_v3_B.INPUT_15_1_1[0];$  $tmp_0[13] = Code_MNLR_v3_B.INPUT_15_1_1[1];$ tmp 0[14] = Code MNLR v3 B.INPUT 15 1 1[2]; $tmp_0[15] = Code_MNLR_v3_B.INPUT_15_1_1[3];$  $tmp_1[4] = 16;$ tmp 0[16] = Code MNLR v3 B.INPUT 21 1 1[0]; $tmp_0[17] = Code_MNLR_v3_B.INPUT_21_1[1];$ tmp\_0[18] = Code\_MNLR\_v3\_B.INPUT\_21\_1\_1[2];  $tmp_0[19] = Code_MNLR_v3_B.INPUT_21_1_1[3];$  $tmp_1[5] = 20;$ tmp 0[20] = Code MNLR v3 B.INPUT 24 1 1[0];tmp 0[21] = Code\_MNLR\_v3\_B.INPUT\_24\_1\_1[1];  $tmp_0[22] = Code_MNLR_v3_B.INPUT_24_1_1[2];$  $tmp_0[23] = Code_MNLR_v3_B.INPUT_24_1_1[3];$  $tmp_1[6] = 24;$  $tmp_0[24] = Code_MNLR_v3_B.INPUT_18_1_1[0];$ tmp\_0[25] = Code\_MNLR\_v3\_B.INPUT\_18\_1\_1[1]; tmp\_0[26] = Code\_MNLR\_v3\_B.INPUT\_18\_1\_1[2]; tmp 0[27] = Code MNLR v3 B.INPUT 18 1 1[3]; $tmp_1[7] = 28;$ tmp\_0[28] = Code\_MNLR\_v3\_B.INPUT\_26\_1\_1[0]; tmp 0[29] = Code MNLR v3 B.INPUT 26 1 1[1]; $tmp_0[30] = Code_MNLR_v3_B.INPUT_26_1_1[2];$  $tmp_0[31] = Code_MNLR_v3_B.INPUT_26_1_1[3];$ tmp 1[8] = 32; tmp\_0[32] = Code\_MNLR\_v3\_B.INPUT\_29\_1\_1[0]; tmp 0[33] = Code MNLR v3 B.INPUT 29 1 1[1];tmp\_0[34] = Code\_MNLR\_v3\_B.INPUT\_29\_1 1[2]; tmp\_0[35] = Code\_MNLR\_v3\_B.INPUT\_29\_1\_1[3]; tmp 1[9] = 36; $tmp_0[36] = Code_MNLR_v3_B.INPUT_31_1_1[0];$  $tmp_0[37] = Code_MNLR_v3_B.INPUT_31_1_1[1];$  $tmp_0[38] = Code_MNLR_v3_B.INPUT_31_1_1[2];$ tmp\_0[39] = Code\_MNLR\_v3\_B.INPUT\_31\_1\_1[3]; tmp 1[10] = 40;tmp\_0[40] = Code\_MNLR\_v3\_B.INPUT\_2\_1\_1[0];  $tmp_0[41] = Code_MNLR_v3_B.INPUT_2_1_1[1];$ tmp 0[42] = Code MNLR v3 B.INPUT 2 1 1[2]; $tmp_0[43] = Code_MNLR_v3_B.INPUT_2_1_1[3];$ tmp 1[11] = 44; $tmp_0[44] = Code_MNLR_v3_B.INPUT_28_1_1[0];$ 

(&Code\_MNLR\_v3\_M->solverInfo);

tmp\_0[45] = Code\_MNLR\_v3\_B.INPUT\_28\_1\_1[1]; tmp 0[46] = Code MNLR v3 B.INPUT 28 1 1[2]; $tmp_0[47] = Code_MNLR_v3_B.INPUT_28_1_1[3];$  $tmp_1[12] = 48;$ tmp 0[48] = Code MNLR v3 B.INPUT 5 1 1[0]; $tmp_0[49] = Code_MNLR_v3_B.INPUT_5_1_1[1];$ tmp 0[50] = Code MNLR v3 B.INPUT 5 1 1[2]; $tmp_0[51] = Code_MNLR_v3_B.INPUT_5_1_1[3];$  $tmp_1[13] = 52;$  $tmp_0[52] = Code_MNLR_v3_B.INPUT_7_1_1[0];$  $tmp_0[53] = Code_MNLR_v3_B.INPUT_7_1_1[1];$ tmp 0[54] = Code MNLR v3 B.INPUT 7 1 1[2];tmp\_0[55] = Code\_MNLR\_v3\_B.INPUT\_7\_1\_1[3];  $tmp_1[14] = 56;$ tmp 0[56] = Code MNLR v3 B.INPUT 9 1 1[0];tmp\_0[57] = Code\_MNLR\_v3\_B.INPUT\_9\_1\_1[1];  $tmp_0[58] = Code_MNLR_v3_B.INPUT_9_1_1[2];$ tmp\_0[59] = Code\_MNLR\_v3\_B.INPUT\_9\_1\_1[3];  $tmp_1[15] = 60;$ tmp 0[60] = Code MNLR v3 B.INPUT 17 1 1[0]; $tmp_0[61] = Code_MNLR_v3_B.INPUT_17_1_1[1];$  $tmp_0[62] = Code_MNLR_v3_B.INPUT_17_1_1[2];$ tmp 0[63] = Code MNLR v3 B.INPUT 17 1 1[3]; $tmp_1[16] = 64;$ tmp\_0[64] = Code\_MNLR\_v3\_B.INPUT\_3\_1\_1[0];  $tmp_0[65] = Code_MNLR_v3_B.INPUT_3_1_1[1];$ tmp\_0[66] = Code\_MNLR\_v3\_B.INPUT\_3\_1\_1[2]; tmp 0[67] = Code MNLR v3 B.INPUT 3 1 1[3]; $tmp_1[17] = 68;$  $tmp_0[68] = Code_MNLR_v3_B.INPUT_4_1_1[0];$  $tmp_0[69] = Code_MNLR_v3_B.INPUT_4_1_1[1];$  $tmp_0[70] = Code_MNLR_v3_B.INPUT_4_1_1[2];$  $tmp_0[71] = Code_MNLR_v3_B.INPUT_4_1_1[3];$  $tmp_1[18] = 72;$ tmp\_0[72] = Code\_MNLR\_v3\_B.INPUT\_6\_1\_1[0]; tmp 0[73] = Code MNLR v3 B.INPUT 6 1 1[1]; $tmp_0[74] = Code_MNLR_v3_B.INPUT_6_1_1[2];$  $tmp_0[75] = Code_MNLR_v3_B.INPUT_6_1_1[3];$ tmp 1[19] = 76; $tmp_0[76] = Code_MNLR_v3_B.INPUT_8_1_1[0];$ tmp\_0[77] = Code\_MNLR\_v3\_B.INPUT\_8\_1\_1[1]; tmp\_0[78] = Code\_MNLR\_v3\_B.INPUT\_8\_1\_1[2]; tmp\_0[79] = Code\_MNLR\_v3\_B.INPUT\_8\_1\_1[3]; tmp 1[20] = 80;tmp 0[80] = Code MNLR v3 B.INPUT 10 1 1[0]; $tmp_0[81] = Code_MNLR_v3_B.INPUT_10_1_1[1];$  $tmp_0[82] = Code_MNLR_v3_B.INPUT_10_1_1[2];$  $tmp_0[83] = Code_MNLR_v3_B.INPUT_10_1_1[3];$  $tmp_1[21] = 84;$  $tmp_0[84] = Code_MNLR_v3_B.INPUT_11_1[0];$ tmp\_0[85] = Code\_MNLR\_v3\_B.INPUT\_11\_1[1]; tmp 0[86] = Code MNLR v3 B.INPUT 11 1 1[2]; $tmp_0[87] = Code_MNLR_v3_B.INPUT_11_1[3];$ tmp 1[22] = 88;tmp 0[88] = Code MNLR v3 B.INPUT 14 1 1[0]; $tmp_0[89] = Code_MNLR_v3_B.INPUT_14_1_1[1];$ tmp\_0[90] = Code\_MNLR\_v3\_B.INPUT\_14\_1\_1[2];  $tmp_0[91] = Code_MNLR_v3_B.INPUT_14_1_1[3];$   $tmp_1[23] = 92;$ tmp 0[92] = Code MNLR v3 B.INPUT 16 1 1[0]; $tmp_0[93] = Code_MNLR_v3_B.INPUT_16_1_1[1];$  $tmp_0[94] = Code_MNLR_v3_B.INPUT_16_1_1[2];$ tmp\_0[95] = Code\_MNLR\_v3\_B.INPUT\_16\_1\_1[3];  $tmp_1[24] = 96;$ tmp 0[96] = Code MNLR v3 B.INPUT 19 1 1[0]; $tmp_0[97] = Code_MNLR_v3_B.INPUT_19_1_1[1];$ tmp\_0[98] = Code\_MNLR\_v3\_B.INPUT\_19\_1\_1[2]; tmp\_0[99] = Code\_MNLR\_v3\_B.INPUT\_19\_1\_1[3];  $tmp_1[25] = 100;$ tmp 0[100] = Code MNLR v3 B.INPUT 20 1 1[0];tmp 0[101] = Code MNLR v3 B.INPUT 20 1 1[1]; $tmp_0[102] = Code_MNLR_v3_B.INPUT_20_1_1[2];$ tmp\_0[103] = Code\_MNLR\_v3\_B.INPUT\_20\_1\_1[3];  $tmp_1[26] = 104;$  $tmp_0[104] = Code_MNLR_v3_B.INPUT_22_1_1[0];$ tmp 0[105] = Code MNLR v3 B.INPUT 22 1 1[1];tmp\_0[106] = Code\_MNLR\_v3\_B.INPUT\_22\_1\_1[2]; tmp 0[107] = Code MNLR v3 B.INPUT 22 1 1[3];tmp 1[27] = 108; $tmp_0[108] = Code_MNLR_v3_B.INPUT_25_1_1[0];$ tmp 0[109] = Code MNLR v3 B.INPUT 25 1 1[1]; $tmp_0[110] = Code_MNLR_v3_B.INPUT_25_1_1[2];$ tmp\_0[111] = Code\_MNLR\_v3\_B.INPUT\_25\_1\_1[3];  $tmp_1[28] = 112;$ tmp\_0[112] = Code\_MNLR\_v3\_B.INPUT\_23\_1\_1[0]; tmp 0[113] = Code MNLR v3 B.INPUT 23 1 1[1];tmp 0[114] = Code\_MNLR\_v3\_B.INPUT\_23\_1\_1[2]; tmp\_0[115] = Code\_MNLR\_v3\_B.INPUT\_23\_1\_1[3]; tmp 1[29] = 116; $tmp_0[116] = Code_MNLR_v3_B.INPUT_27_1_1[0];$  $tmp_0[117] = Code_MNLR_v3_B.INPUT_27_1_1[1];$  $tmp_0[118] = Code_MNLR_v3_B.INPUT_27_1_1[2];$ tmp\_0[119] = Code\_MNLR\_v3\_B.INPUT\_27\_1\_1[3]; tmp 1[30] = 120;tmp\_0[120] = Code\_MNLR\_v3\_B.INPUT\_30\_1\_1[0]; tmp\_0[121] = Code\_MNLR\_v3\_B.INPUT\_30\_1\_1[1]; tmp 0[122] = Code MNLR v3 B.INPUT 30 1 1[2];tmp\_0[123] = Code\_MNLR\_v3\_B.INPUT\_30\_1\_1[3];  $tmp_1[31] = 124;$ tmp\_0[124] = Code\_MNLR\_v3\_B.INPUT\_32\_1\_1[0]; tmp\_0[125] = Code\_MNLR\_v3\_B.INPUT\_32\_1\_1[1]; tmp 0[126] = Code MNLR v3 B.INPUT 32 1 1[2]; $tmp_0[127] = Code_MNLR_v3_B.INPUT_32_1_1[3];$  $tmp_1[32] = 128;$ tmp 0[128] = Code MNLR v3 B.INPUT 33 1 1[0];tmp\_0[129] = Code\_MNLR\_v3\_B.INPUT\_33\_1\_1[1];  $tmp_0[130] = Code_MNLR_v3_B.INPUT_33_1_1[2];$  $tmp_0[131] = Code_MNLR_v3_B.INPUT_33_1_1[3];$  $tmp_1[33] = 132;$ simulationData->mData->mInputValues.mN = 132; simulationData->mData->mInputValues.mX = &tmp 0[0]; simulationData->mData->mInputOffsets.mN = 34; simulationData->mData->mInputOffsets.mX = &tmp 1[0]; simulationData->mData->mDx.mN = 245; simulationData->mData->mDx.mX = &\_rtXdot->Code\_MNLR\_v3Double\_Acting\_Hydra[0];

diagnosticManager = (NeuDiagnosticManager \*)Code\_MNLR\_v3\_DW.STATE\_1\_DiagMgr;

diagnosticTree = neu\_diagnostic\_manager\_get\_initial\_tree(diagnosticManager); tmp 2 = ne simulator method((NeslSimulator \*)Code MNLR v3 DW.STATE 1 Simulator,NESL\_SIM\_DERIVATIVES, simulationData, diagnosticManager); if (tmp 2!=0) { tmp = error buffer is empty(rtmGetErrorStatus(Code MNLR v3 M)); if (tmp) { msg = rtw diagnostics msg(diagnosticTree); rtmSetErrorStatus(Code\_MNLR\_v3\_M, msg); } } /\* End of Derivatives for SimscapeExecutionBlock: '<S954>/STATE\_1' \*/ /\* Derivatives for TransferFcn: '<S103>/Transfer Fcn6' \*/ rtXdot->TransferFcn6 CSTATE = 0.0; \_rtXdot->TransferFcn6\_CSTATE += -30.120474261342469 \* Code MNLR v3 X.TransferFcn6 CSTATE; \_rtXdot->TransferFcn6\_CSTATE += Code\_MNLR\_v3\_B.Sum6\_j; /\* Derivatives for TransferFcn: '<S103>/Transfer Fcn4' \*/ rtXdot->TransferFcn4 CSTATE = 0.0; \_rtXdot->TransferFcn4\_CSTATE += -0.0 \* Code\_MNLR\_v3\_X.TransferFcn4\_CSTATE; \_rtXdot->TransferFcn4\_CSTATE += Code\_MNLR\_v3 B.Sum5 e: /\* Derivatives for TransferFcn: '<S103>/Transfer Fcn5' \*/ \_rtXdot->TransferFcn5\_CSTATE = 0.0; rtXdot->TransferFcn5 CSTATE += -142.85714285714286 \* Code\_MNLR\_v3\_X.TransferFcn5\_CSTATE; \_rtXdot->TransferFcn5\_CSTATE += Code\_MNLR\_v3\_B.TransferFcn4; /\* Model initialize function \*/ void Code MNLR v3 initialize(void) /\* Registration code \*/ { /\* Setup solver object \*/ rtsiSetSimTimeStepPtr(&Code\_MNLR\_v3\_M->solverInfo, &Code MNLR v3 M->Timing.simTimeStep); rtsiSetTPtr(&Code\_MNLR\_v3\_M->solverInfo, &rtmGetTPtr(Code\_MNLR\_v3\_M)); rtsiSetStepSizePtr(&Code MNLR v3 M->solverInfo, &Code MNLR v3 M->Timing.stepSize0); rtsiSetdXPtr(&Code MNLR v3 M->solverInfo, &Code MNLR v3 M->derivs); rtsiSetContStatesPtr(&Code MNLR v3 M->solverInfo, (real T \*\*) &Code\_MNLR\_v3\_M->contStates); rtsiSetNumContStatesPtr(&Code\_MNLR\_v3\_M->solverInfo, &Code MNLR v3 M->Sizes.numContStates); rtsiSetNumPeriodicContStatesPtr(&Code\_MNLR\_v3\_M->solverInfo, &Code MNLR v3 M->Sizes.numPeriodicContStates); rtsiSetPeriodicContStateIndicesPtr(&Code MNLR v3 M->solverInfo, &Code\_MNLR\_v3\_M->periodicContStateIndices); rtsiSetPeriodicContStateRangesPtr(&Code MNLR v3 M->solverInfo, &Code\_MNLR\_v3\_M->periodicContStateRanges); rtsiSetErrorStatusPtr(&Code\_MNLR\_v3\_M->solverInfo, (&rtmGetErrorStatus (Code MNLR v3 M))); rtsiSetSolverMassMatrixIr(&Code\_MNLR\_v3\_M->solverInfo, Code MNLR v3 MassMatrix.ir); rtsiSetSolverMassMatrixJc(&Code MNLR v3 M->solverInfo, Code MNLR v3 MassMatrix.jc); rtsiSetSolverMassMatrixPr(&Code MNLR v3 M->solverInfo, Code\_MNLR\_v3\_MassMatrix.pr); rtsiSetRTModelPtr(&Code\_MNLR\_v3\_M->solverInfo, Code\_MNLR\_v3\_M); ł

```
rtsiSetSimTimeStep(&Code_MNLR_v3_M->solverInfo, MAJOR_TIME_STEP);
Code MNLR v3 M->intgData.x0 = Code MNLR v3 M->odeX0;
Code_MNLR_v3_M->intgData.f0 = Code_MNLR_v3_M->odeF0;
Code_MNLR_v3_M->intgData.x1start = Code_MNLR_v3_M->odeX1START;
Code MNLR v3 M->intgData.f1 = Code MNLR v3 M->odeF1;
Code_MNLR_v3_M->intgData.Delta = Code_MNLR_v3_M->odeDELTA;
Code MNLR v3 M->intgData.E = Code MNLR v3 M->odeE;
Code_MNLR_v3_M->intgData.fac = Code_MNLR_v3_M->odeFAC;
/* initialize */
{
int_T i;
real T *f = Code MNLR v3 M->intgData.fac;
 for (i = 0; i < (int T)(sizeof(Code MNLR v3 M->odeFAC)/sizeof(real T)); i++)
 ł
 f[i] = 1.5e-8;
 }
Code MNLR v3 M->intgData.DFDX = Code MNLR v3 M->odeDFDX;
Code_MNLR_v3_M->intgData.W = Code_MNLR_v3_M->odeW;
Code MNLR v3 M->intgData.pivots = Code MNLR v3 M->odePIVOTS;
Code MNLR v3 M->intgData.xtmp = Code MNLR v3 M->odeXTMP;
Code_MNLR_v3_M->intgData.ztmp = Code_MNLR_v3_M->odeZTMP;
Code MNLR v3 M->intgData.M = Code MNLR v3 M->odeMASSMATRIX M;
Code_MNLR_v3_M->intgData.M1 = Code_MNLR_v3_M->odeMASSMATRIX_M1;
Code_MNLR_v3_M->intgData.xdot = Code_MNLR_v3_M->odeXDOT;
Code MNLR v3 M->intgData.Edot = Code MNLR v3 M->odeEDOT;
Code_MNLR_v3_M->intgData.fminusMxdot = Code_MNLR_v3_M->odeFMXDOT;
Code MNLR v3 M->intgData.isFirstStep = true;
rtsiSetSolverExtrapolationOrder(&Code MNLR v3 M->solverInfo, 4);
rtsiSetSolverNumberNewtonIterations(&Code_MNLR_v3_M->solverInfo, 1);
Code_MNLR_v3_M->contStates = ((X_Code_MNLR_v3_T *) &Code_MNLR_v3_X);
Code_MNLR_v3_M->massMatrixType = ((ssMatrixType)3);
Code_MNLR_v3_M->massMatrixNzMax = (149);
Code MNLR v3 M->massMatrixIr = (Code MNLR v3 MassMatrix.ir);
Code_MNLR_v3_M->massMatrixJc = (Code_MNLR_v3_MassMatrix.jc);
Code MNLR v3 M->massMatrixPr = (Code MNLR v3 MassMatrix.pr);
rtsiSetSolverMassMatrixType(&Code_MNLR_v3_M->solverInfo, (ssMatrixType)3);
rtsiSetSolverMassMatrixNzMax(&Code MNLR v3 M->solverInfo, 149);
rtsiSetSolverData(&Code MNLR v3 M->solverInfo, (void *)
         &Code_MNLR_v3_M->intgData);
rtsiSetIsMinorTimeStepWithModeChange(&Code_MNLR_v3_M->solverInfo, false);
rtsiSetSolverName(&Code_MNLR_v3_M->solverInfo,"ode14x");
rtmSetTPtr(Code_MNLR_v3_M, &Code_MNLR_v3_M->Timing.tArray[0]);
Code MNLR v3 M->Timing.stepSize0 = 0.17482517482517482;
/* Root-level init GlobalMassMatrixPr offset */
 Code MNLR v3 DW.STATE 1 MASS MATRIX PR = 33;/* '<S954>/STATE 1' */
 NeModelParameters modelParameters:
 NeModelParameters modelParameters_0;
 NeslRtpManager *manager;
 NeslRtpManager *manager_0;
 NeslSimulationData *tmp 1;
 NeslSimulator *tmp 0;
 NeuDiagnosticManager *diagnosticManager;
 NeuDiagnosticTree *diagnosticTree;
 NeuDiagnosticTree *diagnosticTree_0;
```

char \*msg; char \*msg 0; real\_T tmp\_2; int32\_T tmp\_3; int T tmp 5; int\_T tmp\_6; int T tmp 7; int T tmp 8; int\_T tmp\_9; boolean T tmp; boolean\_T tmp\_4; /\* Start for SimscapeRtp: '<S952>/RTP 1' \*/ manager 0 = nesl lease rtp manager( "Code\_MNLR\_v3/Subsystem/Solver Configuration1\_1", 0); manager = manager 0;tmp = pointer\_is\_null(manager\_0); if (tmp) { Code MNLR v3 86dd9022 1 gateway(); manager = nesl\_lease\_rtp\_manager( "Code MNLR v3/Subsystem/Solver Configuration 1 ", 0); Code\_MNLR\_v3\_DW.RTP\_1\_RtpManager = (void \*)manager; Code MNLR v3 DW.RTP 1 SetParametersNeeded = true; /\* End of Start for SimscapeRtp: '<S952>/RTP\_1' \*/ /\* Start for TransportDelay: '<Root>/Transport Delay9' \*/ real\_T \*pBuffer = &Code\_MNLR\_v3\_DW.TransportDelay9\_RWORK.TUbufferArea[0]; Code MNLR v3 DW.TransportDelay9 IWORK.Tail = 0; Code MNLR v3 DW.TransportDelay9 IWORK.Head = 0; Code\_MNLR\_v3\_DW.TransportDelay9\_IWORK.Last = 0; Code MNLR v3 DW.TransportDelay9 IWORK.CircularBufSize = 1024; pBuffer[0] = 0.0;pBuffer[1024] = Code\_MNLR\_v3\_M->Timing.t[0]; Code MNLR v3 DW.TransportDelay9 PWORK.TUbufferPtrs[0] = (void \*) &pBuffer [0]; /\* Start for TransportDelay: '<Root>/Transport Delay10' \*/ real T \*pBuffer = &Code MNLR v3 DW.TransportDelay10 RWORK.TUbufferArea[0]; Code\_MNLR\_v3\_DW.TransportDelay10\_IWORK.Tail = 0; Code\_MNLR\_v3\_DW.TransportDelay10\_IWORK.Head = 0; Code\_MNLR\_v3\_DW.TransportDelay10\_IWORK.Last = 0; Code\_MNLR\_v3\_DW.TransportDelay10\_IWORK.CircularBufSize = 1024; pBuffer[0] = 0.0;pBuffer[1024] = Code MNLR v3 M->Timing.t[0]; Code\_MNLR\_v3\_DW.TransportDelay10\_PWORK.TUbufferPtrs[0] = (void \*) &pBuffer[0]; /\* Start for TransportDelay: '<Root>/Transport Delay11' \*/ real\_T \*pBuffer = &Code\_MNLR\_v3\_DW.TransportDelay11\_RWORK.TUbufferArea[0]; Code MNLR v3 DW.TransportDelay11 IWORK.Tail = 0; Code\_MNLR\_v3\_DW.TransportDelay11\_IWORK.Head = 0; Code\_MNLR\_v3\_DW.TransportDelay11\_IWORK.Last = 0; Code MNLR v3 DW.TransportDelay11 IWORK.CircularBufSize = 1024; pBuffer[0] = 0.0; pBuffer[1024] = Code\_MNLR\_v3\_M->Timing.t[0]; Code\_MNLR\_v3\_DW.TransportDelay11\_PWORK.TUbufferPtrs[0] = (void \*)
```
&pBuffer[0];
```

```
}
/* Start for TransportDelay: '<Root>/Transport Delay12' */
{
 real T *pBuffer = &Code MNLR v3 DW.TransportDelay12 RWORK.TUbufferArea[0];
 Code_MNLR_v3_DW.TransportDelay12_IWORK.Tail = 0;
 Code MNLR v3 DW.TransportDelay12 IWORK.Head = 0;
 Code MNLR v3 DW.TransportDelay12 IWORK.Last = 0;
 Code_MNLR_v3_DW.TransportDelay12_IWORK.CircularBufSize = 1024;
 pBuffer[0] = 0.0;
pBuffer[1024] = Code_MNLR_v3_M->Timing.t[0];
 Code MNLR v3 DW.TransportDelay12 PWORK.TUbufferPtrs[0] = (void *)
  &pBuffer[0];
/* Start for TransportDelay: '<Root>/Transport Delay13' */
{
 real_T *pBuffer = &Code_MNLR_v3_DW.TransportDelay13_RWORK.TUbufferArea[0];
 Code MNLR v3 DW.TransportDelay13 IWORK.Tail = 0;
 Code_MNLR_v3_DW.TransportDelay13_IWORK.Head = 0;
 Code_MNLR_v3_DW.TransportDelav13 IWORK.Last = 0:
 Code MNLR v3 DW.TransportDelay13 IWORK.CircularBufSize = 1024;
 pBuffer[0] = 0.0;
 pBuffer[1024] = Code MNLR v3 M->Timing.t[0];
 Code_MNLR_v3_DW.TransportDelay13_PWORK.TUbufferPtrs[0] = (void *)
  &pBuffer[0];
/* Start for TransportDelay: '<Root>/Transport Delay14' */
real T*pBuffer = &Code MNLR v3 DW.TransportDelav14 RWORK.TUbufferArea[0];
 Code_MNLR_v3_DW.TransportDelay14_IWORK.Tail = 0;
 Code MNLR v3 DW.TransportDelay14 IWORK.Head = 0;
 Code_MNLR_v3_DW.TransportDelay14_IWORK.Last = 0;
 Code MNLR v3 DW.TransportDelay14 IWORK.CircularBufSize = 1024;
 pBuffer[0] = 0.0;
 pBuffer[1024] = Code_MNLR_v3_M->Timing.t[0];
 Code MNLR v3 DW.TransportDelay14 PWORK.TUbufferPtrs[0] = (void *)
  &pBuffer[0];
/* Start for TransportDelay: '<Root>/Transport Delay15' */
{
 real_T *pBuffer = &Code_MNLR_v3_DW.TransportDelay15_RWORK.TUbufferArea[0];
Code_MNLR_v3_DW.TransportDelay15_IWORK.Tail = 0;
 Code_MNLR_v3_DW.TransportDelay15_IWORK.Head = 0;
 Code MNLR v3 DW.TransportDelay15 IWORK.Last = 0;
 Code MNLR v3 DW.TransportDelay15 IWORK.CircularBufSize = 1024;
 pBuffer[0] = 0.0;
 pBuffer[1024] = Code MNLR v3 M->Timing.t[0];
Code_MNLR_v3_DW.TransportDelay15_PWORK.TUbufferPtrs[0] = (void *)
  &pBuffer[0];
}
/* Start for TransportDelay: '<Root>/Transport Delay1' */
{
real_T *pBuffer = &Code_MNLR_v3_DW.TransportDelay1_RWORK.TUbufferArea[0];
 Code MNLR v3 DW.TransportDelay1 IWORK.Tail = 0;
 Code MNLR v3 DW.TransportDelay1 IWORK.Head = 0;
 Code_MNLR_v3_DW.TransportDelay1_IWORK.Last = 0;
 Code_MNLR_v3_DW.TransportDelay1_IWORK.CircularBufSize = 1024;
 pBuffer[0] = 0.0;
```

pBuffer[1024] = Code\_MNLR\_v3\_M->Timing.t[0]; Code\_MNLR\_v3\_DW.TransportDelay1\_PWORK.TUbufferPtrs[0] = (void \*) &pBuffer [0]; /\* Start for TransportDelay: '<Root>/Transport Delay16' \*/ real T \*pBuffer = &Code MNLR v3 DW.TransportDelay16 RWORK.TUbufferArea[0]; Code MNLR v3 DW.TransportDelay16 IWORK.Tail = 0; Code\_MNLR\_v3\_DW.TransportDelay16\_IWORK.Head = 0; Code MNLR v3 DW.TransportDelay16 IWORK.Last = 0; Code\_MNLR\_v3\_DW.TransportDelay16\_IWORK.CircularBufSize = 1024; pBuffer[0] = 0.0;pBuffer[1024] = Code\_MNLR\_v3 M->Timing.t[0]: Code\_MNLR\_v3\_DW.TransportDelay16\_PWORK.TUbufferPtrs[0] = (void \*) &pBuffer[0]; /\* Start for TransportDelay: '<Root>/Transport Delay2' \*/ { real\_T \*pBuffer = &Code\_MNLR\_v3\_DW.TransportDelay2\_RWORK.TUbufferArea[0]; Code MNLR v3 DW.TransportDelay2 IWORK.Tail = 0; Code MNLR v3 DW.TransportDelay2 IWORK.Head = 0; Code\_MNLR\_v3\_DW.TransportDelay2\_IWORK.Last = 0; Code MNLR v3 DW.TransportDelay2 IWORK.CircularBufSize = 1024; pBuffer[0] = 0.0;pBuffer[1024] = Code\_MNLR\_v3\_M->Timing.t[0]; Code MNLR v3 DW.TransportDelay2 PWORK.TUbufferPtrs[0] = (void \*) &pBuffer [0]; } /\* Start for TransportDelay: '<Root>/Transport Delay3' \*/ real T \*pBuffer = &Code MNLR v3 DW.TransportDelay3 RWORK.TUbufferArea[0]; Code\_MNLR\_v3\_DW.TransportDelay3\_IWORK.Tail = 0; Code\_MNLR\_v3\_DW.TransportDelay3\_IWORK.Head = 0; Code MNLR v3 DW.TransportDelay3 IWORK.Last = 0; Code\_MNLR\_v3\_DW.TransportDelay3\_IWORK.CircularBufSize = 1024; pBuffer[0] = 0.0;pBuffer[1024] = Code MNLR v3 M->Timing.t[0]; Code MNLR v3 DW.TransportDelay3 PWORK.TUbufferPtrs[0] = (void \*) &pBuffer [0]; } /\* Start for TransportDelay: '<Root>/Transport Delay4' \*/ ł real\_T \*pBuffer = &Code\_MNLR\_v3\_DW.TransportDelay4\_RWORK.TUbufferArea[0]; Code MNLR v3 DW.TransportDelay4 IWORK.Tail = 0; Code MNLR v3 DW.TransportDelay4 IWORK.Head = 0; Code\_MNLR\_v3\_DW.TransportDelay4\_IWORK.Last = 0; Code MNLR v3 DW.TransportDelay4 IWORK.CircularBufSize = 1024; pBuffer[0] = 0.0;pBuffer[1024] = Code\_MNLR\_v3\_M->Timing.t[0]; Code\_MNLR\_v3\_DW.TransportDelay4\_PWORK.TUbufferPtrs[0] = (void \*) &pBuffer [0]; /\* Start for TransportDelay: '<Root>/Transport Delay5' \*/ real T \*pBuffer = &Code MNLR v3 DW.TransportDelay5 RWORK.TUbufferArea[0]; Code\_MNLR\_v3\_DW.TransportDelay5\_IWORK.Tail = 0; Code\_MNLR\_v3\_DW.TransportDelay5\_IWORK.Head = 0; Code\_MNLR\_v3\_DW.TransportDelay5\_IWORK.Last = 0;

Code\_MNLR\_v3\_DW.TransportDelay5\_IWORK.CircularBufSize = 1024; pBuffer[0] = 0.0;pBuffer[1024] = Code\_MNLR\_v3\_M->Timing.t[0]; Code\_MNLR\_v3\_DW.TransportDelay5\_PWORK.TUbufferPtrs[0] = (void \*) &pBuffer [0]; /\* Start for TransportDelay: '<Root>/Transport Delay6' \*/ real\_T \*pBuffer = &Code\_MNLR\_v3\_DW.TransportDelay6\_RWORK.TUbufferArea[0]; Code MNLR v3 DW.TransportDelay6 IWORK.Tail = 0; Code\_MNLR\_v3\_DW.TransportDelay6\_IWORK.Head = 0; Code MNLR v3 DW.TransportDelay6 IWORK.Last = 0; Code MNLR v3 DW.TransportDelay6 IWORK.CircularBufSize = 1024; pBuffer[0] = 0.0;pBuffer[1024] = Code MNLR v3 M->Timing.t[0]; Code\_MNLR\_v3\_DW.TransportDelay6\_PWORK.TUbufferPtrs[0] = (void \*) &pBuffer [0]; } /\* Start for TransportDelay: '<Root>/Transport Delay7' \*/ real T \*pBuffer = &Code MNLR v3 DW.TransportDelay7 RWORK.TUbufferArea[0]; Code\_MNLR\_v3\_DW.TransportDelay7\_IWORK.Tail = 0; Code MNLR v3 DW.TransportDelay7 IWORK.Head = 0; Code\_MNLR\_v3\_DW.TransportDelay7\_IWORK.Last = 0; Code\_MNLR\_v3\_DW.TransportDelay7\_IWORK.CircularBufSize = 1024; pBuffer[0] = 0.0;pBuffer[1024] = Code\_MNLR\_v3\_M->Timing.t[0]; Code MNLR v3 DW.TransportDelay7 PWORK.TUbufferPtrs[0] = (void \*) &pBuffer [0]: ł /\* Start for TransportDelay: '<Root>/Transport Delay8' \*/ ł real\_T \*pBuffer = &Code\_MNLR\_v3\_DW.TransportDelay8\_RWORK.TUbufferArea[0]; Code MNLR v3 DW.TransportDelay8 IWORK.Tail = 0; Code\_MNLR\_v3\_DW.TransportDelay8\_IWORK.Head = 0; Code MNLR v3 DW.TransportDelay8 IWORK.Last = 0; Code MNLR v3 DW.TransportDelay8 IWORK.CircularBufSize = 1024; pBuffer[0] = 0.0;pBuffer[1024] = Code MNLR v3 M->Timing.t[0]; Code\_MNLR\_v3\_DW.TransportDelay8\_PWORK.TUbufferPtrs[0] = (void \*) &pBuffer [0]; ł /\* Start for SimscapeExecutionBlock: '<S954>/STATE\_1' \*/ tmp 0 = nesl lease simulator( "Code MNLR v3/Subsystem/Solver Configuration1 1", 0, 0); Code\_MNLR\_v3\_DW.STATE\_1\_Simulator = (void \*)tmp\_0; tmp = pointer is null(Code MNLR v3 DW.STATE 1 Simulator); if (tmp) { Code\_MNLR\_v3\_86dd9022\_1\_gateway(); tmp 0 = nesl lease simulator( "Code\_MNLR\_v3/Subsystem/Solver Configuration1\_1", 0, 0); Code MNLR v3 DW.STATE 1 Simulator = (void \*)tmp 0; tmp 1 = nesl create simulation data(); Code MNLR v3 DW.STATE 1 SimData = (void \*)tmp 1; diagnosticManager = rtw\_create\_diagnostics(); Code\_MNLR\_v3\_DW.STATE\_1\_DiagMgr = (void \*)diagnosticManager; modelParameters.mSolverType = NE\_SOLVER\_TYPE\_DAE;

```
modelParameters.mSolverAbsTol = 0.001;
modelParameters.mSolverRelTol = 0.001;
modelParameters.mSolverModifyAbsTol = NE_MODIFY_ABS_TOL_NO;
modelParameters.mStartTime = 0.0;
modelParameters.mLoadInitialState = false:
modelParameters.mUseSimState = false;
modelParameters.mLinTrimCompile = false;
modelParameters.mLoggingMode = SSC LOGGING NONE;
modelParameters.mRTWModifiedTimeStamp = 6.49380572E+8;
tmp 2 = 0.001;
modelParameters.mSolverTolerance = tmp_2;
tmp 2 = 0.17482517482517482;
modelParameters.mFixedStepSize = tmp 2;
tmp = false;
modelParameters.mVariableStepSolver = tmp;
tmp = false;
modelParameters.mIsUsingODEN = tmp;
modelParameters.mZcDisabled = true;
diagnosticManager = (NeuDiagnosticManager *)Code_MNLR_v3_DW.STATE_1_DiagMgr;
diagnosticTree = neu diagnostic manager get initial tree(diagnosticManager);
tmp 3 = nesl initialize simulator((NeslSimulator *))
 Code_MNLR_v3_DW.STATE_1_Simulator, &modelParameters, diagnosticManager);
if (tmp 3!=0) {
 tmp = error_buffer_is_empty(rtmGetErrorStatus(Code_MNLR_v3_M));
 if (tmp) {
  msg = rtw diagnostics msg(diagnosticTree);
 rtmSetErrorStatus(Code_MNLR_v3_M, msg);
 }
}
/* End of Start for SimscapeExecutionBlock: '<S954>/STATE_1' */
/* Start for SimscapeExecutionBlock: '<S954>/OUTPUT 1 0' */
tmp_0 = nesl_lease_simulator(
 "Code_MNLR_v3/Subsystem/Solver Configuration1_1", 1, 0);
Code MNLR v3 DW.OUTPUT 1 0 Simulator = (void *)tmp 0;
tmp = pointer_is_null(Code_MNLR_v3_DW.OUTPUT_1_0_Simulator);
if (tmp) {
 Code_MNLR_v3_86dd9022_1_gateway();
 tmp 0 = nesl lease simulator(
  "Code MNLR v3/Subsystem/Solver Configuration 1 1", 1, 0);
 Code_MNLR_v3_DW.OUTPUT_1_0_Simulator = (void *)tmp_0;
}
tmp_1 = nesl_create_simulation_data();
Code_MNLR_v3_DW.OUTPUT_1_0_SimData = (void *)tmp_1;
diagnosticManager = rtw create diagnostics();
Code MNLR v3 DW.OUTPUT 1 0 DiagMgr = (void *)diagnosticManager;
modelParameters_0.mSolverType = NE_SOLVER_TYPE_DAE;
modelParameters 0.mSolverAbsTol = 0.001;
modelParameters_0.mSolverRelTol = 0.001;
modelParameters 0.mSolverModifyAbsTol = NE MODIFY ABS TOL NO;
modelParameters 0.mStartTime = 0.0;
modelParameters_0.mLoadInitialState = false;
modelParameters 0.mUseSimState = false;
modelParameters 0.mLinTrimCompile = false;
modelParameters_0.mLoggingMode = SSC_LOGGING NONE:
modelParameters 0.mRTWModifiedTimeStamp = 6.49380572E+8;
tmp_2 = 0.001;
modelParameters_0.mSolverTolerance = tmp_2;
tmp_2 = 0.17482517482517482;
```

modelParameters\_0.mFixedStepSize = tmp\_2; tmp = false;modelParameters\_0.mVariableStepSolver = tmp; tmp = false;modelParameters 0.mIsUsingODEN = tmp: modelParameters\_0.mZcDisabled = true; diagnosticManager = (NeuDiagnosticManager \*) Code\_MNLR\_v3\_DW.OUTPUT\_1\_0\_DiagMgr; diagnosticTree\_0 = neu\_diagnostic\_manager\_get\_initial\_tree(diagnosticManager); tmp 3 = nesl initialize simulator((NeslSimulator \*)) Code\_MNLR\_v3\_DW.OUTPUT\_1\_0\_Simulator, &modelParameters\_0, diagnosticManager); if (tmp 3!=0) { tmp = error\_buffer\_is\_empty(rtmGetErrorStatus(Code\_MNLR\_v3\_M)); if (tmp) { msg\_0 = rtw\_diagnostics\_msg(diagnosticTree\_0); rtmSetErrorStatus(Code\_MNLR\_v3\_M, msg\_0); } } /\* End of Start for SimscapeExecutionBlock: '<S954>/OUTPUT 1 0' \*/ /\* InitializeConditions for Integrator: '<S214>/Integrator' \*/ Code\_MNLR\_v3\_X.Integrator\_CSTATE = 0.0; /\* InitializeConditions for Integrator: '<S209>/Filter' \*/ Code\_MNLR\_v3\_X.Filter\_CSTATE = 0.0; /\* InitializeConditions for Integrator: '<S886>/Integrator' \*/ Code MNLR v3 X.Integrator CSTATE p = 0.0; /\* InitializeConditions for Integrator: '<S881>/Filter' \*/ Code MNLR v3 X.Filter CSTATE p = 0.0; /\* InitializeConditions for Integrator: '<S262>/Integrator' \*/ Code\_MNLR\_v3\_X.Integrator\_CSTATE\_n = 0.0; /\* InitializeConditions for Integrator: '<S257>/Filter' \*/ Code\_MNLR\_v3\_X.Filter\_CSTATE\_i = 0.0; /\* InitializeConditions for Integrator: '<S310>/Integrator' \*/ Code MNLR v3 X.Integrator CSTATE i = 0.0; /\* InitializeConditions for Integrator: '<S305>/Filter' \*/ Code\_MNLR\_v3\_X.Filter\_CSTATE\_k = 0.0; /\* InitializeConditions for Integrator: '<S358>/Integrator' \*/ Code MNLR v3 X.Integrator CSTATE m = 0.0; /\* InitializeConditions for Integrator: '<S353>/Filter' \*/ Code\_MNLR\_v3\_X.Filter\_CSTATE\_h = 0.0; /\* InitializeConditions for Integrator: '<S502>/Integrator' \*/ Code\_MNLR\_v3\_X.Integrator\_CSTATE\_l = 0.0; /\* InitializeConditions for Integrator: '<S497>/Filter' \*/ Code MNLR v3 X.Filter CSTATE a = 0.0; /\* InitializeConditions for Integrator: '<S406>/Integrator' \*/ Code\_MNLR\_v3\_X.Integrator\_CSTATE\_ma = 0.0; /\* InitializeConditions for Integrator: '<S401>/Filter' \*/ Code\_MNLR\_v3\_X.Filter\_CSTATE\_g = 0.0; /\* InitializeConditions for Integrator: '<S550>/Integrator' \*/ Code\_MNLR\_v3\_X.Integrator\_CSTATE\_js = 0.0; /\* InitializeConditions for Integrator: '<S545>/Filter' \*/ Code MNLR v3 X.Filter CSTATE gy = 0.0; /\* InitializeConditions for Integrator: '<S454>/Integrator' \*/ Code MNLR v3 X.Integrator CSTATE a = 0.0; /\* InitializeConditions for Integrator: '<S449>/Filter' \*/ Code\_MNLR\_v3\_X.Filter\_CSTATE\_ay = 0.0; /\* InitializeConditions for Integrator: '<S166>/Integrator' \*/ Code\_MNLR\_v3\_X.Integrator\_CSTATE\_f = 0.0;

/\* InitializeConditions for Integrator: '<S161>/Filter' \*/ Code MNLR v3 X.Filter CSTATE n = 0.0; /\* InitializeConditions for Integrator: '<S598>/Integrator' \*/ Code\_MNLR\_v3\_X.Integrator\_CSTATE\_i = 0.0; /\* InitializeConditions for Integrator: '<S593>/Filter' \*/ Code\_MNLR\_v3\_X.Filter\_CSTATE\_o = 0.0; /\* InitializeConditions for Integrator: '<S646>/Integrator' \*/ Code\_MNLR\_v3\_X.Integrator\_CSTATE\_k = 0.0; /\* InitializeConditions for Integrator: '<S641>/Filter' \*/ Code MNLR v3 X.Filter CSTATE pb = 0.0; /\* InitializeConditions for TransferFcn: '<S103>/Transfer Fcn7' \*/ Code MNLR v3 X.TransferFcn7 CSTATE = 0.0; /\* InitializeConditions for Integrator: '<S742>/Integrator' \*/ Code\_MNLR\_v3\_X.Integrator\_CSTATE\_it = 0.0; /\* InitializeConditions for Integrator: '<S737>/Filter' \*/ Code\_MNLR\_v3\_X.Filter\_CSTATE\_d = 0.0; /\* InitializeConditions for Integrator: '<S694>/Integrator' \*/ Code MNLR v3 X.Integrator CSTATE kg = 0.0; /\* InitializeConditions for Integrator: '<S689>/Filter' \*/ Code MNLR v3 X.Filter CSTATE i4 = 0.0; /\* InitializeConditions for Integrator: '<S790>/Integrator' \*/ Code\_MNLR\_v3\_X.Integrator\_CSTATE\_j2 = 0.0; /\* InitializeConditions for Integrator: '<S785>/Filter' \*/ Code MNLR v3 X.Filter CSTATE i = 0.0; /\* InitializeConditions for Integrator: '<S838>/Integrator' \*/ Code MNLR v3 X.Integrator CSTATE ft = 0.0; /\* InitializeConditions for Integrator: '<S833>/Filter' \*/ Code MNLR v3 X.Filter CSTATE ku = 0.0; /\* InitializeConditions for SimscapeExecutionBlock: '<S954>/STATE 1' \*/ tmp = false;tmp 4 =false; if (tmp\_4 || tmp) { tmp\_3 = strcmp(rtsiGetSolverName(&Code\_MNLR\_v3\_M->solverInfo), "daessc"); tmp 5 = strcmp(rtsiGetSolverName(&Code MNLR v3 M->solverInfo), "ode14x"); tmp\_6 = strcmp(rtsiGetSolverName(&Code\_MNLR\_v3\_M->solverInfo), "ode15s"); tmp 7 = strcmp(rtsiGetSolverName(&Code MNLR v3 M->solverInfo), "ode1be"); tmp\_8 = strcmp(rtsiGetSolverName(&Code\_MNLR\_v3\_M->solverInfo), "ode23t"); tmp\_9 = strcmp(rtsiGetSolverName(&Code\_MNLR\_v3\_M->solverInfo), "odeN"); if ((boolean T)((tmp  $3 \neq 0)$  & (tmp  $5 \neq 0$ ) & (tmp  $6 \neq 0$ ) & (tmp  $7 \neq 0$ ) &  $(tmp_8 != 0)$  &  $(tmp_9 != 0))) {$ rtmSetErrorStatus(Code MNLR v3 M, "Detected inconsistent solvers in the model reference hierarchy. Model built with

ode14x requires one of {daessc, ode14x, ode15s, ode1be, ode23t, odeN} solvers to run. Use one of the required solvers in the top model.");

} Code\_MNLR\_v3\_X.TransferFcn6\_CSTATE = 0.0; Code\_MNLR\_v3\_X.TransferFcn4\_CSTATE = 0.0; Code\_MNLR\_v3\_X.TransferFcn5\_CSTATE = 0.0; { static int\_T modelMassMatrixIr[149] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128,

}

129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 278, 279, 280 }; static int\_T modelMassMatrixJc[282] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 1.0, 1.0 }; (void) memcpy(Code\_MNLR\_v3\_MassMatrix.ir, modelMassMatrixIr, 149\*sizeof(int\_T)); (void) memcpy(Code\_MNLR\_v3\_MassMatrix.jc, modelMassMatrixJc, 282\*sizeof(int\_T)); (void) memcpy(Code\_MNLR\_v3\_MassMatrix.pr, modelMassMatrixPr, 149\*sizeof(real T)); } } void Code\_MNLR\_v3\_terminate(void) neu\_destroy\_diagnostic\_manager((NeuDiagnosticManager \*) Code\_MNLR\_v3\_DW.STATE\_1\_DiagMgr); nesl destroy simulation data((NeslSimulationData \*) Code\_MNLR\_v3\_DW.STATE\_1\_SimData); nesl\_erase\_simulator("Code\_MNLR\_v3/Subsystem/Solver Configuration1\_1"); nesl\_destroy\_registry(); neu\_destroy\_diagnostic\_manager((NeuDiagnosticManager \*) Code MNLR v3 DW.OUTPUT 1 0 DiagMgr); nesl\_destroy\_simulation\_data((NeslSimulationData \*) Code MNLR v3 DW.OUTPUT 1 0 SimData); nesl erase simulator("Code MNLR v3/Subsystem/Solver Configuration1 1"); nesl\_destroy\_registry();

}

ł

223