Table of Contents

==========

==========

```
ПРОДВИНУТАЯ СИМУЛЯЦИЯ ДИНАМИКИ ТЕХНИЧЕСКОГО ДОЛГА В РАЗРАБОТКЕ ПО
================================================================
Теория становится практикой: от уравнений к визуализации в MATLAB
Три сценария разработки: выбор стратегии и её экономические последствия
================================================================
```

```
clear all; close all; clc;
format long g;
set(0, 'DefaultFigureWindowStyle', 'normal');
warning('off', 'all');
```

========== ПАРАМЕТРЫ СИСТЕМЫ ==========

```
fprintf('================================================\n');
fprintf('ЗАПУСК ПРОДВИНУТОЙ СИМУЛЯЦИИ ТЕХНИЧЕСКОГО ДОЛГА\n');
fprintf('================================================\n\n');

% Общие начальные условия (одинаковые для всех сценариев)
fprintf('Общие начальные условия:\n');
fprintf('  Чистый проект: D₀ = 0\n');
fprintf('  Начальная ценность: V₀ = 0\n');
fprintf('  Скорость команды: R = 10 ед./неделю\n\n');

% Параметры симуляции
R = 10;                         % Скорость команды (единиц ценности в неделю)
T_max = 100;                    % Максимальное время симуляции (недель)
dt = 0.02;                      % Шаг времени для точного интегрирования
t = 0:dt:T_max;                 % Вектор времени

% Параметры среды
sigma_noise = 0.15;             % Уровень шума в системе
```

```
market_volatility = 0.1;     % Волатильность рыночных условий
innovation_rate = 0.05;      % Скорость появления новых технологий

============================================
ЗАПУСК ПРОДВИНУТОЙ СИМУЛЯЦИИ ТЕХНИЧЕСКОГО ДОЛГА
============================================

Общие начальные условия:
  Чистый проект: D₀ = 0
  Начальная ценность: V₀ = 0
  Скорость команды: R = 10 ед./неделю
```

================ СЦЕНАРИЙ 1: "БЕГ С ПРЕПЯТСТВИЯМИ" ==============

```
fprintf('------------------------------------------------\n');
fprintf('СЦЕНАРИЙ 1: "БЕГ С ПРЕПЯТСТВИЯМИ"\n');
fprintf('------------------------------------------------\n');
fprintf('Стратегия: Постоянная скорость, игнорирование долга\n');
fprintf('Параметры: k = 0.3 (среднее качество), γ = 0.02\n\n');

% Параметры сценария
k1 = 0.3;                    % Среднее качество (порождает долг)
gamma1 = 0.02;               % Коэффициент влияния долга на скорость

% Инициализация массивов
V1 = zeros(size(t));         % Ценность продукта
D1 = zeros(size(t));         % Технический долг
Q1 = zeros(size(t));         % Качество кода
E1 = zeros(size(t));         % Эффективность команды
P1 = zeros(size(t));         % Потенциал системы

% Начальные условия
V1(1) = 0.1;                 % Малая начальная ценность
D1(1) = 0;                   % Начальный долг = 0
Q1(1) = 1.0;                 % Начальное качество (идеальное)
E1(1) = 1.0;                 % Начальная эффективность команды

% Динамические переменные для сложных эффектов
omega1 = zeros(size(t));     % Фаза сложных колебаний
theta1 = zeros(size(t));     % Угол для спиральных эффектов
noise1 = sigma_noise * randn(size(t)); % Внешний шум

% Генерация событий (критические точки)
events_time = [15, 35, 60, 85];
events_magnitude = [0.3, 0.5, 0.4, 0.6];

% Основной цикл симуляции
fprintf('Расчет динамики системы...\n');
for i = 2:length(t)
    % ========== СЛОЖНАЯ ДИНАМИКА ДОЛГА ==========
```

```
% Нелинейный рост с эффектом насыщения
D_base = k1 * R * (1 - tanh(D1(i-1)/20));

% Эффект накопления (долг порождает долг)
D_compound = gamma1 * D1(i-1)^1.5 * exp(-D1(i-1)/50);

% Влияние качества на генерацию долга
D_quality = (1 - Q1(i-1)) * R * 0.2;

% Суммарный рост долга
D_growth = D_base + D_compound + D_quality;

% ========== СЛОЖНАЯ ДИНАМИКА ЦЕННОСТИ ==========
% Базовая скорость с учетом эффективности
V_base = R * E1(i-1);

% Влияние долга на скорость (нелинейное)
V_debt_effect = 1 - gamma1 * D1(i-1) * (1 + 0.3*sin(omega1(i-1)));

% Влияние качества на ценность
V_quality_effect = Q1(i-1)^2;

% Рыночная волатильность
V_market = market_volatility * sin(0.1*t(i) + noise1(i));

% Суммарная скорость роста ценности
V_growth = V_base * V_debt_effect * V_quality_effect + V_market;

% ========== ДИНАМИКА КАЧЕСТВА ==========
% Естественная деградация качества
Q_degradation = 0.01 * (1 - Q1(i-1));

% Влияние долга на качество
Q_debt_effect = 0.05 * D1(i-1) / (1 + D1(i-1));

% Инвестиции в качество (обратно пропорциональны скорости)
Q_investment = 0.02 * (1 - E1(i-1));

% Изменение качества
Q_change = -Q_degradation - Q_debt_effect + Q_investment;

% ========== ДИНАМИКА ЭФФЕКТИВНОСТИ ==========
% Влияние долга на эффективность (нелинейное)
E_debt_effect = 1 / (1 + 0.1 * D1(i-1)^1.2);

% Усталость команды (накапливается со временем)
E_fatigue = 0.001 * t(i) * exp(-0.01*t(i));

% Обучение и адаптация
E_learning = 0.005 * (1 - exp(-0.05*t(i)));

% Изменение эффективности
E_change = (E_debt_effect - E_fatigue + E_learning) * E1(i-1);
```

```
    % ========== СЛОЖНЫЕ ЭФФЕКТЫ И ОСЦИЛЛЯЦИИ ==========
    % Фазовые переменные
    omega1(i) = omega1(i-1) + dt * (0.1 + 0.05*sin(D1(i-1)/10));
    theta1(i) = theta1(i-1) + dt * 0.15;

    % Спиральные эффекты
    spiral_effect = 0.01 * sin(theta1(i)) * exp(-0.02*t(i));

    % ========== КРИТИЧЕСКИЕ СОБЫТИЯ ==========
    event_effect = 0;
    for ev = 1:length(events_time)
        if abs(t(i) - events_time(ev)) < 0.5
            event_effect = -events_magnitude(ev) * R;
            fprintf('Событие в t=%.1f: потеря %.1f ед. скорости\n', ...
                    t(i), abs(event_effect));
        end
    end

    % ========== ИНТЕГРАЦИЯ УРАВНЕНИЙ ==========
    D1(i) = D1(i-1) + dt * D_growth;
    V1(i) = V1(i-1) + dt * (V_growth + event_effect + spiral_effect);
    Q1(i) = max(0.1, min(1.0, Q1(i-1) + dt * Q_change));
    E1(i) = max(0.3, min(1.2, E1(i-1) + dt * E_change));

    % Потенциал системы
    P1(i) = V1(i) - gamma1 * D1(i)^2 / (1 + exp(-0.1*(t(i)-30)));
end

------------------------------------------------
СЦЕНАРИЙ 1: "БЕГ С ПРЕПЯТСТВИЯМИ"
------------------------------------------------
Стратегия: Постоянная скорость, игнорирование долга
Параметры: k = 0.3 (среднее качество), γ = 0.02

Расчет динамики системы...
Событие в t=14.5: потеря 3.0 ед. скорости
Событие в t=14.5: потеря 3.0 ед. скорости
Событие в t=14.6: потеря 3.0 ед. скорости
Событие в t=14.6: потеря 3.0 ед. скорости
Событие в t=14.6: потеря 3.0 ед. скорости
Событие в t=14.6: потеря 3.0 ед. скорости
Событие в t=14.6: потеря 3.0 ед. скорости
Событие в t=14.7: потеря 3.0 ед. скорости
Событие в t=14.7: потеря 3.0 ед. скорости
Событие в t=14.7: потеря 3.0 ед. скорости
Событие в t=14.7: потеря 3.0 ед. скорости
Событие в t=14.7: потеря 3.0 ед. скорости
Событие в t=14.8: потеря 3.0 ед. скорости
Событие в t=14.8: потеря 3.0 ед. скорости
Событие в t=14.8: потеря 3.0 ед. скорости
Событие в t=14.8: потеря 3.0 ед. скорости
Событие в t=14.8: потеря 3.0 ед. скорости
Событие в t=14.9: потеря 3.0 ед. скорости
Событие в t=14.9: потеря 3.0 ед. скорости
```

```
Событие в t=14.9: потеря 3.0 ед. скорости
Событие в t=14.9: потеря 3.0 ед. скорости
Событие в t=14.9: потеря 3.0 ед. скорости
Событие в t=15.0: потеря 3.0 ед. скорости
Событие в t=15.0: потеря 3.0 ед. скорости
Событие в t=15.0: потеря 3.0 ед. скорости
Событие в t=15.0: потеря 3.0 ед. скорости
Событие в t=15.0: потеря 3.0 ед. скорости
Событие в t=15.1: потеря 3.0 ед. скорости
Событие в t=15.1: потеря 3.0 ед. скорости
Событие в t=15.1: потеря 3.0 ед. скорости
Событие в t=15.1: потеря 3.0 ед. скорости
Событие в t=15.1: потеря 3.0 ед. скорости
Событие в t=15.2: потеря 3.0 ед. скорости
Событие в t=15.2: потеря 3.0 ед. скорости
Событие в t=15.2: потеря 3.0 ед. скорости
Событие в t=15.2: потеря 3.0 ед. скорости
Событие в t=15.2: потеря 3.0 ед. скорости
Событие в t=15.3: потеря 3.0 ед. скорости
Событие в t=15.3: потеря 3.0 ед. скорости
Событие в t=15.3: потеря 3.0 ед. скорости
Событие в t=15.3: потеря 3.0 ед. скорости
Событие в t=15.3: потеря 3.0 ед. скорости
Событие в t=15.4: потеря 3.0 ед. скорости
Событие в t=15.4: потеря 3.0 ед. скорости
Событие в t=15.4: потеря 3.0 ед. скорости
Событие в t=15.4: потеря 3.0 ед. скорости
Событие в t=15.4: потеря 3.0 ед. скорости
Событие в t=15.5: потеря 3.0 ед. скорости
Событие в t=15.5: потеря 3.0 ед. скорости
Событие в t=34.5: потеря 5.0 ед. скорости
Событие в t=34.5: потеря 5.0 ед. скорости
Событие в t=34.6: потеря 5.0 ед. скорости
Событие в t=34.6: потеря 5.0 ед. скорости
Событие в t=34.6: потеря 5.0 ед. скорости
Событие в t=34.6: потеря 5.0 ед. скорости
Событие в t=34.6: потеря 5.0 ед. скорости
Событие в t=34.7: потеря 5.0 ед. скорости
Событие в t=34.7: потеря 5.0 ед. скорости
Событие в t=34.7: потеря 5.0 ед. скорости
Событие в t=34.7: потеря 5.0 ед. скорости
Событие в t=34.7: потеря 5.0 ед. скорости
Событие в t=34.8: потеря 5.0 ед. скорости
Событие в t=34.8: потеря 5.0 ед. скорости
Событие в t=34.8: потеря 5.0 ед. скорости
Событие в t=34.8: потеря 5.0 ед. скорости
Событие в t=34.8: потеря 5.0 ед. скорости
Событие в t=34.9: потеря 5.0 ед. скорости
Событие в t=34.9: потеря 5.0 ед. скорости
Событие в t=34.9: потеря 5.0 ед. скорости
Событие в t=34.9: потеря 5.0 ед. скорости
Событие в t=34.9: потеря 5.0 ед. скорости
Событие в t=35.0: потеря 5.0 ед. скорости
Событие в t=35.0: потеря 5.0 ед. скорости
```

```
Событие в t=35.0: потеря 5.0 ед. скорости
Событие в t=35.0: потеря 5.0 ед. скорости
Событие в t=35.0: потеря 5.0 ед. скорости
Событие в t=35.1: потеря 5.0 ед. скорости
Событие в t=35.1: потеря 5.0 ед. скорости
Событие в t=35.1: потеря 5.0 ед. скорости
Событие в t=35.1: потеря 5.0 ед. скорости
Событие в t=35.1: потеря 5.0 ед. скорости
Событие в t=35.2: потеря 5.0 ед. скорости
Событие в t=35.2: потеря 5.0 ед. скорости
Событие в t=35.2: потеря 5.0 ед. скорости
Событие в t=35.2: потеря 5.0 ед. скорости
Событие в t=35.2: потеря 5.0 ед. скорости
Событие в t=35.3: потеря 5.0 ед. скорости
Событие в t=35.3: потеря 5.0 ед. скорости
Событие в t=35.3: потеря 5.0 ед. скорости
Событие в t=35.3: потеря 5.0 ед. скорости
Событие в t=35.3: потеря 5.0 ед. скорости
Событие в t=35.4: потеря 5.0 ед. скорости
Событие в t=35.4: потеря 5.0 ед. скорости
Событие в t=35.4: потеря 5.0 ед. скорости
Событие в t=35.4: потеря 5.0 ед. скорости
Событие в t=35.4: потеря 5.0 ед. скорости
Событие в t=35.5: потеря 5.0 ед. скорости
Событие в t=35.5: потеря 5.0 ед. скорости
Событие в t=59.5: потеря 4.0 ед. скорости
Событие в t=59.5: потеря 4.0 ед. скорости
Событие в t=59.6: потеря 4.0 ед. скорости
Событие в t=59.6: потеря 4.0 ед. скорости
Событие в t=59.6: потеря 4.0 ед. скорости
Событие в t=59.6: потеря 4.0 ед. скорости
Событие в t=59.6: потеря 4.0 ед. скорости
Событие в t=59.7: потеря 4.0 ед. скорости
Событие в t=59.7: потеря 4.0 ед. скорости
Событие в t=59.7: потеря 4.0 ед. скорости
Событие в t=59.7: потеря 4.0 ед. скорости
Событие в t=59.7: потеря 4.0 ед. скорости
Событие в t=59.8: потеря 4.0 ед. скорости
Событие в t=59.8: потеря 4.0 ед. скорости
Событие в t=59.8: потеря 4.0 ед. скорости
Событие в t=59.8: потеря 4.0 ед. скорости
Событие в t=59.8: потеря 4.0 ед. скорости
Событие в t=59.9: потеря 4.0 ед. скорости
Событие в t=59.9: потеря 4.0 ед. скорости
Событие в t=59.9: потеря 4.0 ед. скорости
Событие в t=59.9: потеря 4.0 ед. скорости
Событие в t=59.9: потеря 4.0 ед. скорости
Событие в t=60.0: потеря 4.0 ед. скорости
Событие в t=60.0: потеря 4.0 ед. скорости
Событие в t=60.0: потеря 4.0 ед. скорости
Событие в t=60.0: потеря 4.0 ед. скорости
Событие в t=60.0: потеря 4.0 ед. скорости
Событие в t=60.1: потеря 4.0 ед. скорости
Событие в t=60.1: потеря 4.0 ед. скорости
```

```
Событие в t=60.1: потеря 4.0 ед. скорости
Событие в t=60.1: потеря 4.0 ед. скорости
Событие в t=60.1: потеря 4.0 ед. скорости
Событие в t=60.2: потеря 4.0 ед. скорости
Событие в t=60.2: потеря 4.0 ед. скорости
Событие в t=60.2: потеря 4.0 ед. скорости
Событие в t=60.2: потеря 4.0 ед. скорости
Событие в t=60.2: потеря 4.0 ед. скорости
Событие в t=60.3: потеря 4.0 ед. скорости
Событие в t=60.3: потеря 4.0 ед. скорости
Событие в t=60.3: потеря 4.0 ед. скорости
Событие в t=60.3: потеря 4.0 ед. скорости
Событие в t=60.3: потеря 4.0 ед. скорости
Событие в t=60.4: потеря 4.0 ед. скорости
Событие в t=60.4: потеря 4.0 ед. скорости
Событие в t=60.4: потеря 4.0 ед. скорости
Событие в t=60.4: потеря 4.0 ед. скорости
Событие в t=60.4: потеря 4.0 ед. скорости
Событие в t=60.5: потеря 4.0 ед. скорости
Событие в t=60.5: потеря 4.0 ед. скорости
Событие в t=84.5: потеря 6.0 ед. скорости
Событие в t=84.5: потеря 6.0 ед. скорости
Событие в t=84.6: потеря 6.0 ед. скорости
Событие в t=84.6: потеря 6.0 ед. скорости
Событие в t=84.6: потеря 6.0 ед. скорости
Событие в t=84.6: потеря 6.0 ед. скорости
Событие в t=84.6: потеря 6.0 ед. скорости
Событие в t=84.7: потеря 6.0 ед. скорости
Событие в t=84.7: потеря 6.0 ед. скорости
Событие в t=84.7: потеря 6.0 ед. скорости
Событие в t=84.7: потеря 6.0 ед. скорости
Событие в t=84.7: потеря 6.0 ед. скорости
Событие в t=84.8: потеря 6.0 ед. скорости
Событие в t=84.8: потеря 6.0 ед. скорости
Событие в t=84.8: потеря 6.0 ед. скорости
Событие в t=84.8: потеря 6.0 ед. скорости
Событие в t=84.8: потеря 6.0 ед. скорости
Событие в t=84.9: потеря 6.0 ед. скорости
Событие в t=84.9: потеря 6.0 ед. скорости
Событие в t=84.9: потеря 6.0 ед. скорости
Событие в t=84.9: потеря 6.0 ед. скорости
Событие в t=84.9: потеря 6.0 ед. скорости
Событие в t=85.0: потеря 6.0 ед. скорости
Событие в t=85.0: потеря 6.0 ед. скорости
Событие в t=85.0: потеря 6.0 ед. скорости
Событие в t=85.0: потеря 6.0 ед. скорости
Событие в t=85.0: потеря 6.0 ед. скорости
Событие в t=85.1: потеря 6.0 ед. скорости
Событие в t=85.1: потеря 6.0 ед. скорости
Событие в t=85.1: потеря 6.0 ед. скорости
Событие в t=85.1: потеря 6.0 ед. скорости
Событие в t=85.1: потеря 6.0 ед. скорости
Событие в t=85.2: потеря 6.0 ед. скорости
Событие в t=85.2: потеря 6.0 ед. скорости
```

```
Событие в t=85.2: потеря 6.0 ед. скорости
Событие в t=85.2: потеря 6.0 ед. скорости
Событие в t=85.2: потеря 6.0 ед. скорости
Событие в t=85.3: потеря 6.0 ед. скорости
Событие в t=85.3: потеря 6.0 ед. скорости
Событие в t=85.3: потеря 6.0 ед. скорости
Событие в t=85.3: потеря 6.0 ед. скорости
Событие в t=85.3: потеря 6.0 ед. скорости
Событие в t=85.4: потеря 6.0 ед. скорости
Событие в t=85.4: потеря 6.0 ед. скорости
Событие в t=85.4: потеря 6.0 ед. скорости
Событие в t=85.4: потеря 6.0 ед. скорости
Событие в t=85.4: потеря 6.0 ед. скорости
Событие в t=85.5: потеря 6.0 ед. скорости
Событие в t=85.5: потеря 6.0 ед. скорости
```

========== ПОДГОТОВКА 4D/5D ВИЗУАЛИЗАЦИЙ ==========

```
fprintf('\nПодготовка продвинутых визуализаций...\n');

% Создание 4D сетки (x, y, z, интенсивность)
[X_4D, Y_4D, Z_4D] = meshgrid(linspace(0, T_max, 25), ...
                              linspace(min(V1), max(V1), 25), ...
                              linspace(0, max(D1), 15));

% Расчет 4D скалярного поля (потенциал системы в 4D)
Phi_4D = zeros(size(X_4D));
for idx = 1:numel(X_4D)
    t_val = X_4D(idx);
    v_val = Y_4D(idx);
    d_val = Z_4D(idx);

    % Сложный 4D потенциал
    Phi_4D(idx) = v_val - gamma1 * d_val^2 * ...
                  (1 + 0.2*sin(0.2*t_val)) * ...
                  exp(-d_val/(20 + 5*sin(0.1*t_val)));
end

% Расчет градиента 4D поля
[FX, FY, FZ] = gradient(Phi_4D, ...
                        (T_max/24), (max(V1)-min(V1))/24, max(D1)/14);

% Нормализация для визуализации
F_mag = sqrt(FX.^2 + FY.^2 + FZ.^2);
FX_norm = FX ./ (F_mag + eps);
FY_norm = FY ./ (F_mag + eps);
FZ_norm = FZ ./ (F_mag + eps);

% Создание 5D тензорного поля (добавляем время как 5-е измерение)
T_5D = linspace(0, T_max, 8);
Tensor_5D = zeros([size(X_4D), length(T_5D)]);
```

```
for t_idx = 1:length(T_5D)
    for idx = 1:numel(X_4D)
        t_val = X_4D(idx);
        v_val = Y_4D(idx);
        d_val = Z_4D(idx);
        tau = T_5D(t_idx);

        % 5D тензорное поле с временной эволюцией
        Tensor_5D(idx + (t_idx-1)*numel(X_4D)) = ...
            exp(-0.1*abs(t_val - tau)) * ...
            (v_val - gamma1*d_val^2/(1 + exp(-0.05*(tau-20))));
    end
end
```

Подготовка продвинутых визуализаций...

========== СОЗДАНИЕ ГЛАВНОГО ИНТЕРФЕЙСА ==========

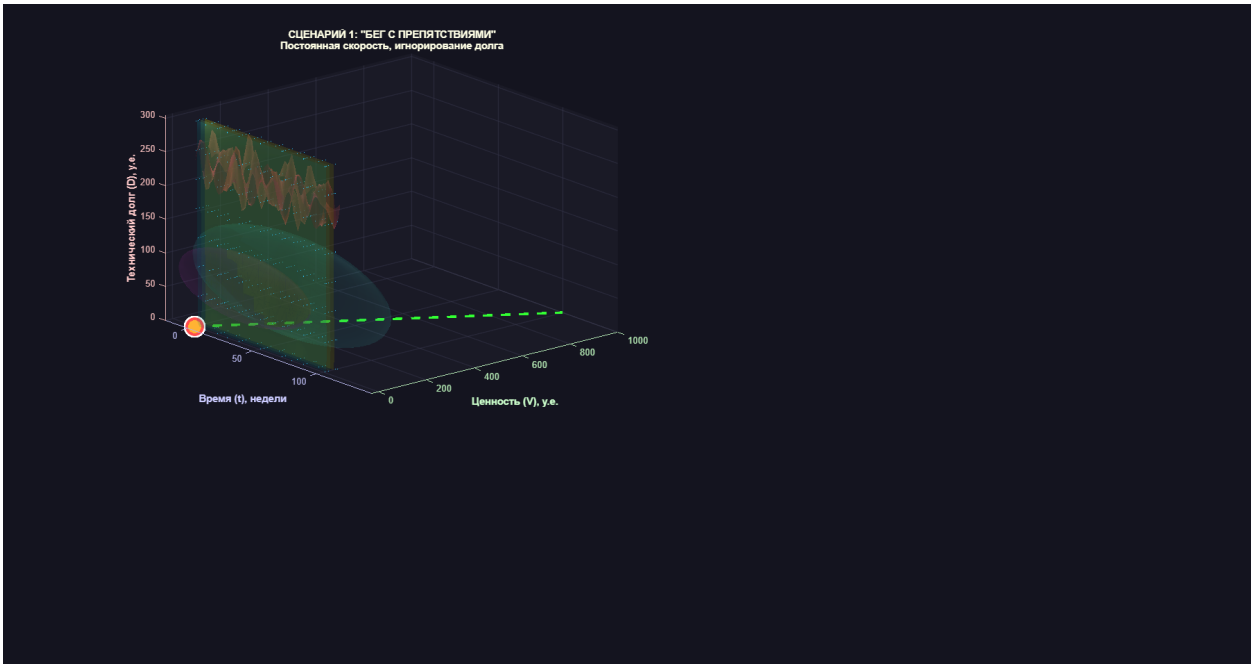
```
fprintf('Создание интерфейса визуализации...\n');

main_fig = figure('Position', [50, 50, 1800, 950], ...
                  'Color', [0.08 0.08 0.12], ...
                  'Name', 'Продвинутая симуляция технического долга', ...
                  'NumberTitle', 'off', ...
                  'MenuBar', 'none', ...
                  'ToolBar', 'none');

% ============ ОСНОВНАЯ 3D АНИМАЦИЯ ============
ax_main = subplot(3, 4, [1 2 5 6]);
set(ax_main, 'Color', [0.1 0.1 0.15], ...
             'GridColor', [0.3 0.3 0.4], ...
             'GridAlpha', 0.3, ...
             'XColor', [0.7 0.7 0.9], ...
             'YColor', [0.7 0.9 0.7], ...
             'ZColor', [0.9 0.7 0.7]);

hold on;

% 1. Изоповерхности 4D поля (разные уровни)
iso_levels = linspace(min(Phi_4D(:)), max(Phi_4D(:)), 5);
iso_colors = jet(length(iso_levels));
iso_handles = [];

for iso_idx = 1:length(iso_levels)
    try
        iso_handles(iso_idx) = patch(isosurface(X_4D, Y_4D, Z_4D, Phi_4D,
iso_levels(iso_idx)), ...
                                     'FaceColor', iso_colors(iso_idx,:), ...
                                     'EdgeColor', 'none', ...
```

```
                                    'FaceAlpha', 0.08 + 0.04*iso_idx, ...
                                    'FaceLighting', 'gouraud', ...
                                    'SpecularStrength', 0.3);
    catch
        continue;
    end
end

% 2. Векторное поле градиента
quiv_scale = 0.6;
quiv_handle = quiver3(X_4D(1:2:end, 1:2:end, 1:2:end), ...
                      Y_4D(1:2:end, 1:2:end, 1:2:end), ...
                      Z_4D(1:2:end, 1:2:end, 1:2:end), ...
                      FX_norm(1:2:end, 1:2:end, 1:2:end) * quiv_scale, ...
                      FY_norm(1:2:end, 1:2:end, 1:2:end) * quiv_scale, ...
                      FZ_norm(1:2:end, 1:2:end, 1:2:end) * quiv_scale, ...
                      0, 'Color', [0.2 0.8 1], 'LineWidth', 0.8, ...
                      'MaxHeadSize', 0.8, 'AutoScale', 'off');

% 3. Главная траектория
traj_handle = plot3(t(1), V1(1), D1(1), ...
                    'Color', [1 1 1], 'LineWidth', 3.5, ...
                    'LineStyle', '-');

% 4. Текущая точка с эффектом свечения
point_outer = plot3(t(1), V1(1), D1(1), 'o', ...
                    'MarkerSize', 22, 'MarkerFaceColor', [1 0.3 0.3], ...
                    'MarkerEdgeColor', [1 1 1], 'LineWidth', 2);

point_inner = plot3(t(1), V1(1), D1(1), 'o', ...
                    'MarkerSize', 14, 'MarkerFaceColor', [1 0.7 0.2], ...
                    'MarkerEdgeColor', 'none');

% 5. Динамическая "стена долга"
[X_wall, Y_wall] = meshgrid(linspace(0, T_max, 40), ...
                            linspace(min(V1), max(V1)*1.2, 30));

% Создание сложной динамической поверхности
Z_wall_base = 0.8 * max(D1);
wall_wave1 = 0.15 * sin(2*pi*X_wall/25) .* cos(2*pi*Y_wall/max(V1));
wall_wave2 = 0.1 * sin(2*pi*X_wall/15 + pi/4) .* sin(2*pi*Y_wall/
(max(V1)*0.8));
Z_wall = Z_wall_base * (1 + wall_wave1 + wall_wave2);

wall_handle = surf(X_wall, Y_wall, Z_wall, ...
                   'FaceColor', [1 0.4 0.4], ...
                   'EdgeColor', 'none', ...
                   'FaceAlpha', 0.25, ...
                   'FaceLighting', 'phong', ...
                   'SpecularColorReflectance', 0.5, ...
                   'SpecularExponent', 25, ...
                   'DiffuseStrength', 0.6);

% 6. Идеальная траектория (без долга)
```

```
ideal_V = R * t;
ideal_handle = plot3(t, ideal_V, zeros(size(t)), ...
                     'Color', [0.2 1 0.2], 'LineWidth', 2.5, ...
                     'LineStyle', '--');

% 7. Фазовый портрет на плоскости



% 8. Области притяжения/отталкивания
theta_sphere = linspace(0, 2*pi, 25);
phi_sphere = linspace(0, pi, 25);
[Theta, Phi] = meshgrid(theta_sphere, phi_sphere);

% Создание нескольких сфер
sphere_centers = [T_max/3, max(V1)/2, max(D1)/4;
                  T_max*2/3, max(V1)*0.8, max(D1)/3];
sphere_radii = [max(D1)/6, max(D1)/4];
sphere_colors = [[0.8 0.2 0.8]; [0.2 0.8 0.8]];

for sph = 1:size(sphere_centers, 1)
    X_sph = sphere_centers(sph,1) + sphere_radii(sph) * sin(Phi) .*
cos(Theta);
    Y_sph = sphere_centers(sph,2) + sphere_radii(sph) * sin(Phi) .*
sin(Theta);
    Z_sph = sphere_centers(sph,3) + sphere_radii(sph) * cos(Phi);

    surf(X_sph, Y_sph, Z_sph, ...
        'FaceColor', sphere_colors(sph,:), ...
        'EdgeColor', 'none', ...
        'FaceAlpha', 0.08, ...
        'FaceLighting', 'gouraud');
end

% Настройка освещения
light('Position', [T_max, max(V1), max(D1)], ...
      'Style', 'local', 'Color', [1 1 0.9]);
light('Position', [0, min(V1), max(D1)], ...
      'Style', 'local', 'Color', [0.8 0.8 1]);
light('Position', [T_max/2, max(V1), 0], ...
      'Style', 'local', 'Color', [0.9 1 0.8]);
lighting gouraud;
material([0.4 0.6 0.4 10 0.5]);

% Настройка осей и заголовка
xlabel('Время (t), недели', 'FontSize', 12, 'FontWeight', 'bold', ...
       'Color', [0.8 0.8 1]);
ylabel('Ценность (V), у.е.', 'FontSize', 12, 'FontWeight', 'bold', ...
       'Color', [0.8 1 0.8]);
zlabel('Технический долг (D), у.е.', 'FontSize', 12, 'FontWeight', 'bold',
...
       'Color', [1 0.8 0.8]);
title({'СЦЕНАРИЙ 1: "БЕГ С ПРЕПЯТСТВИЯМИ"', ...
       'Постоянная скорость, игнорирование долга'}, ...
```

```
        'FontSize', 14, 'FontWeight', 'bold', 'Color', [1 1 0.9]);

grid on;
view(50, 25);
axis tight;
rotate3d on;
set(gca, 'FontSize', 10);

% Цветовая карта
colormap(jet);
```

Создание интерфейса визуализации...



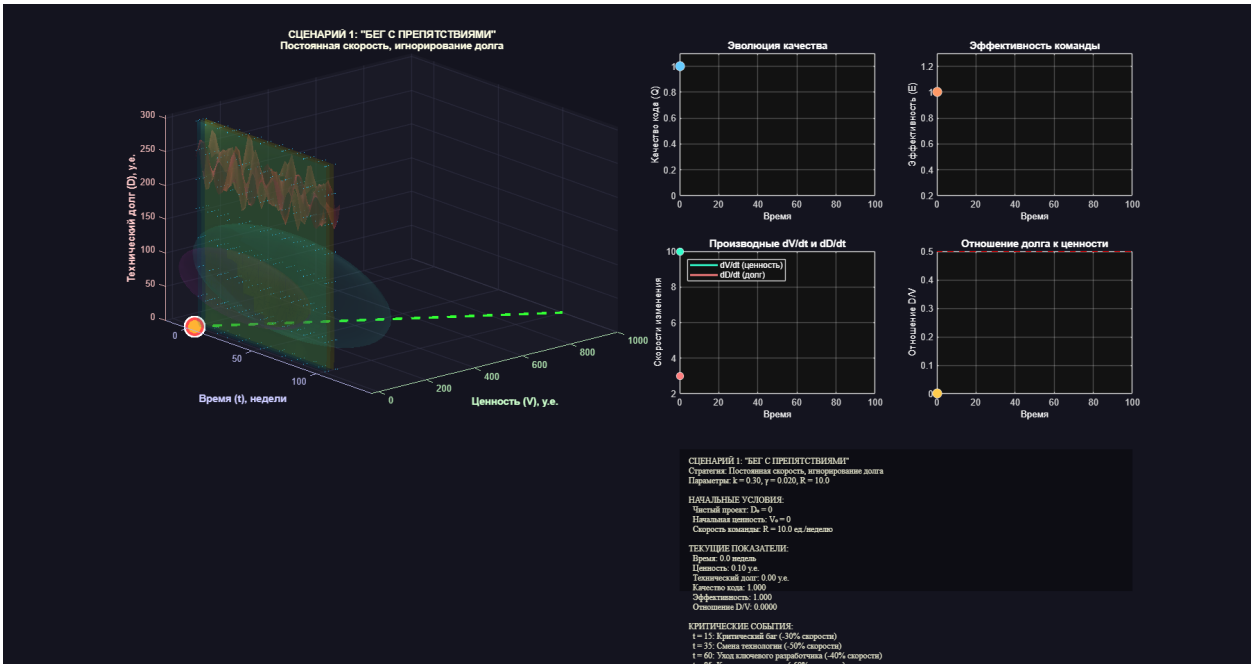
========== ДОПОЛНИТЕЛЬНЫЕ ГРАФИКИ ==========

```
% График качества кода
ax_quality = subplot(3, 4, 3);
set(ax_quality, 'Color', [0.12 0.12 0.16], ...
                'XColor', [0.7 0.7 0.9], ...
                'YColor', [0.7 0.9 0.7]);

quality_handle = plot(t(1), Q1(1), 'Color', [0.4 0.8 1], 'LineWidth', 2.5);
hold on;
quality_point = plot(t(1), Q1(1), 'o', 'MarkerSize', 10, ...
                    'MarkerFaceColor', [0.4 0.8 1], ...
                    'MarkerEdgeColor', [1 1 1]);
grid on;
xlabel('Время', 'FontSize', 10, 'Color', 'white');
ylabel('Качество кода (Q)', 'FontSize', 10, 'Color', 'white');
title('Эволюция качества', 'FontSize', 11, 'Color', 'white', 'FontWeight',
```

```
'bold');
xlim([0 T_max]);
ylim([0 1.1]);

% График эффективности команды
ax_efficiency = subplot(3, 4, 4);
set(ax_efficiency, 'Color', [0.12 0.12 0.16], ...
                   'XColor', [0.7 0.7 0.9], ...
                   'YColor', [0.9 0.7 0.7]);

efficiency_handle = plot(t(1), E1(1), 'Color', [1 0.6 0.4], 'LineWidth',
2.5);
hold on;
efficiency_point = plot(t(1), E1(1), 'o', 'MarkerSize', 10, ...
                        'MarkerFaceColor', [1 0.6 0.4], ...
                        'MarkerEdgeColor', [1 1 1]);
grid on;
xlabel('Время', 'FontSize', 10, 'Color', 'white');
ylabel('Эффективность (E)', 'FontSize', 10, 'Color', 'white');
title('Эффективность команды', 'FontSize', 11, 'Color', 'white',
'FontWeight', 'bold');
xlim([0 T_max]);
ylim([0.2 1.3]);

% График производных
ax_derivatives = subplot(3, 4, 7);
set(ax_derivatives, 'Color', [0.12 0.12 0.16], ...
                    'XColor', [0.7 0.7 0.9], ...
                    'YColor', [0.9 0.9 0.7]);

dVdt = gradient(V1, dt);
dDdt = gradient(D1, dt);

dV_handle = plot(t(1), dVdt(1), 'Color', [0.2 1 0.8], 'LineWidth', 2);
hold on;
dD_handle = plot(t(1), dDdt(1), 'Color', [1 0.5 0.5], 'LineWidth', 2);

dV_point = plot(t(1), dVdt(1), 'o', 'MarkerSize', 8, ...
                'MarkerFaceColor', [0.2 1 0.8], 'MarkerEdgeColor', 'white');
dD_point = plot(t(1), dDdt(1), 'o', 'MarkerSize', 8, ...
                'MarkerFaceColor', [1 0.5 0.5], 'MarkerEdgeColor', 'white');

grid on;
xlabel('Время', 'FontSize', 10, 'Color', 'white');
ylabel('Скорости изменения', 'FontSize', 10, 'Color', 'white');
title('Производные dV/dt и dD/dt', 'FontSize', 11, 'Color', 'white',
'FontWeight', 'bold');
legend('dV/dt (ценность)', 'dD/dt (долг)', 'TextColor', 'white', 'Location',
'northwest');
xlim([0 T_max]);

% График отношения D/V
ax_ratio = subplot(3, 4, 8);
set(ax_ratio, 'Color', [0.12 0.12 0.16], ...
```

```
              'XColor', [0.7 0.7 0.9], ...
              'YColor', [1 0.8 0.5]);

ratio = D1 ./ (V1 + eps);
ratio_handle = plot(t(1), ratio(1), 'Color', [1 0.8 0.3], 'LineWidth', 2.5);
hold on;
ratio_point = plot(t(1), ratio(1), 'o', 'MarkerSize', 10, ...
                   'MarkerFaceColor', [1 0.8 0.3], ...
                   'MarkerEdgeColor', 'white');

% Критический уровень
critical_level = 0.5;
plot([0 T_max], [critical_level critical_level], 'r--', 'LineWidth', 1.5);

grid on;
xlabel('Время', 'FontSize', 10, 'Color', 'white');
ylabel('Отношение D/V', 'FontSize', 10, 'Color', 'white');
title('Отношение долга к ценности', 'FontSize', 11, 'Color', 'white',
'FontWeight', 'bold');
xlim([0 T_max]);

% Информационная панель
ax_info = subplot(3, 4, [11 12]);
set(ax_info, 'Color', [0.05 0.05 0.08], ...
             'XColor', 'none', 'YColor', 'none', ...
             'Box', 'on');

info_text = {sprintf('СЦЕНАРИЙ 1: "БЕГ С ПРЕПЯТСТВИЯМИ"');
             sprintf('Стратегия: Постоянная скорость, игнорирование долга');
             sprintf('Параметры: k = %.2f, γ = %.3f, R = %.1f', k1, gamma1,
R);
             sprintf(' ');
             sprintf('НАЧАЛЬНЫЕ УСЛОВИЯ:');
             sprintf('  Чистый проект: D₀ = 0');
             sprintf('  Начальная ценность: V₀ = 0');
             sprintf('  Скорость команды: R = %.1f ед./неделю', R);
             sprintf(' ');
             sprintf('ТЕКУЩИЕ ПОКАЗАТЕЛИ:');
             sprintf('  Время: %.1f недель', t(1));
             sprintf('  Ценность: %.2f у.е.', V1(1));
             sprintf('  Технический долг: %.2f у.е.', D1(1));
             sprintf('  Качество кода: %.3f', Q1(1));
             sprintf('  Эффективность: %.3f', E1(1));
             sprintf('  Отношение D/V: %.4f', ratio(1));
             sprintf(' ');
             sprintf('КРИТИЧЕСКИЕ СОБЫТИЯ:');
             sprintf('  t = 15: Критический баг (-30%% скорости)');
             sprintf('  t = 35: Смена технологии (-50%% скорости)');
             sprintf('  t = 60: Уход ключевого разработчика (-40%%
скорости)');
             sprintf('  t = 85: Кризис легаси-кода (-60%% скорости)')};

info_handle = text(0.02, 0.95, info_text, ...
                   'FontSize', 9, 'Color', [0.9 0.9 0.8], ...
```

```
                'VerticalAlignment', 'top', ...
                'FontName', 'Monospaced', ...
                'Interpreter', 'none');

axis([0 1 0 1]);
```



========== ПРОДВИНУТАЯ АНИМАЦИЯ ==========

```
fprintf('\nЗапуск продвинутой анимации...\n');
pause(3);

% Настройка скорости анимации
frames_total = min(800, floor(length(t)/5));
skip_factor = floor(length(t)/frames_total);

% Массив для эффектов
trail_length = 20;
trail_x = zeros(1, trail_length);
trail_y = zeros(1, trail_length);
trail_z = zeros(1, trail_length);

% Счетчик кадров
frame_counter = 0;

% Основной цикл анимации
for i = 1:skip_factor:length(t)
    frame_counter = frame_counter + 1;
    current_idx = i;
```

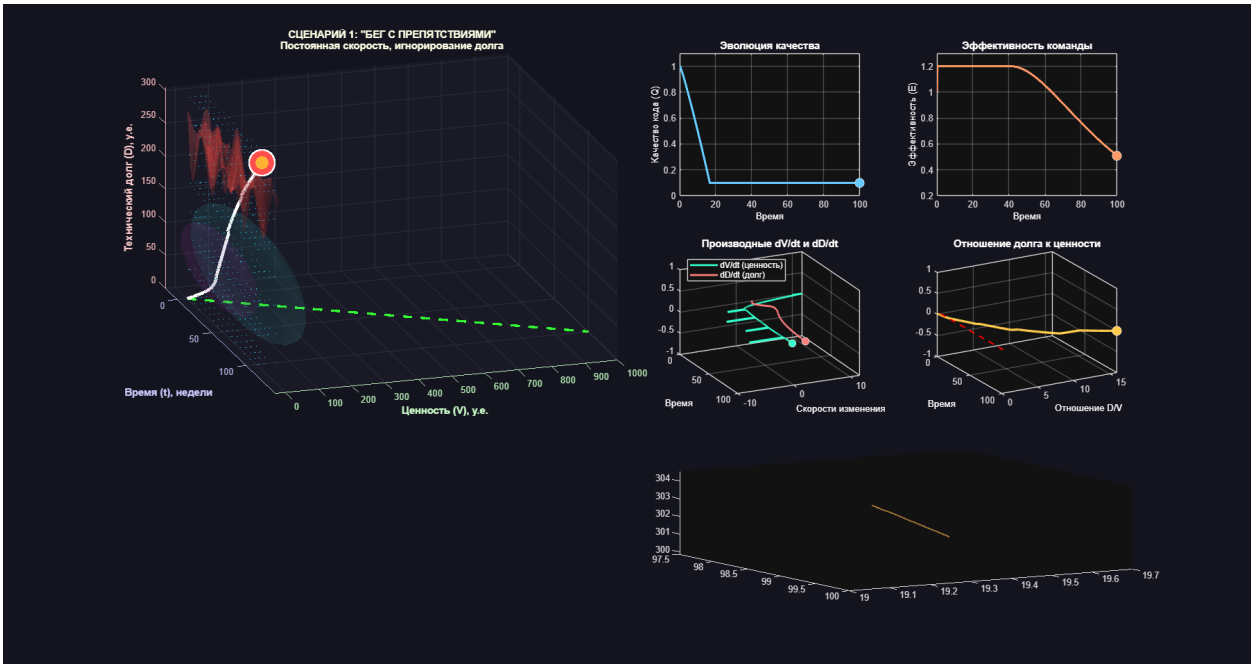
```
    % ========== ОБНОВЛЕНИЕ ОСНОВНОЙ ТРАЕКТОРИИ ==========
    set(traj_handle, 'XData', t(1:current_idx), ...
                     'YData', V1(1:current_idx), ...
                     'ZData', D1(1:current_idx));

    % ========== ЭФФЕКТ ШЛЕЙФА ==========
    trail_x = [trail_x(2:end), t(current_idx)];
    trail_y = [trail_y(2:end), V1(current_idx)];
    trail_z = [trail_z(2:end), D1(current_idx)];

    % Отображение шлейфа
    if exist('trail_handle', 'var')
        delete(trail_handle);
    end
    trail_handle = plot3(trail_x, trail_y, trail_z, ...
                         'Color', [1 0.7 0.3 0.6], ...
                         'LineWidth', 1.5, ...
                         'LineStyle', '-');

    % ========== ОБНОВЛЕНИЕ ТОЧКИ С ЭФФЕКТАМИ ==========
    % Пульсация точки
    pulse_size = 20 + 8 * sin(frame_counter/10);
    set(point_outer, 'XData', t(current_idx), ...
                     'YData', V1(current_idx), ...
                     'ZData', D1(current_idx), ...
                     'MarkerSize', pulse_size);

    set(point_inner, 'XData', t(current_idx), ...
                     'YData', V1(current_idx), ...
                     'ZData', D1(current_idx));

    % ========== ДИНАМИЧЕСКАЯ СТЕНА ДОЛГА ==========
    % Движение волн на стене
    wall_speed = 0.05;
    wall_wave1 = 0.15 * sin(2*pi*(X_wall/25 + wall_speed*frame_counter/100));
    wall_wave2 = 0.1 * sin(2*pi*(X_wall/15 + pi/4 + wall_speed*frame_counter/
80));
    Z_wall_dynamic = Z_wall_base * (1 + wall_wave1 .* cos(2*pi*Y_wall/
max(V1)) + wall_wave2);

    set(wall_handle, 'ZData', Z_wall_dynamic);

    % Эффект мерцания при приближении к стене
    if D1(current_idx) > 0.6 * max(D1)
        wall_alpha = 0.25 + 0.15 * sin(frame_counter/5);
        set(wall_handle, 'FaceAlpha', wall_alpha, ...
                         'FaceColor', [1 0.3 0.3]);
    else
        set(wall_handle, 'FaceAlpha', 0.25, ...
                         'FaceColor', [1 0.4 0.4]);
    end

    % ========== ДИНАМИЧЕСКИЕ ИЗОПОВЕРХНОСТИ ==========
    % Плавное изменение изоповерхностей
```

```
    if mod(frame_counter, 30) == 0
        for iso_idx = 1:length(iso_handles)
            if ishandle(iso_handles(iso_idx))
                delete(iso_handles(iso_idx));
            end
        end

        % Новые изоповерхности с динамическими уровнями
        iso_shift = 0.1 * sin(frame_counter/50);
        iso_levels_dynamic = iso_levels * (1 + iso_shift);

        iso_handles = [];
        for iso_idx = 1:length(iso_levels_dynamic)
            try
                iso_handles(iso_idx) = patch(isosurface(X_4D, Y_4D, Z_4D,
Phi_4D, iso_levels_dynamic(iso_idx)), ...
                                             'FaceColor',
iso_colors(iso_idx,:), ...
                                             'EdgeColor', 'none', ...
                                             'FaceAlpha', 0.08 +
0.04*iso_idx, ...
                                             'FaceLighting', 'gouraud');
            catch
                continue;
            end
        end
    end

    % ========== ДИНАМИЧЕСКИЙ ВЗГЛЯД КАМЕРЫ ==========
    % Плавное движение камеры
    if frame_counter < frames_total/3
        view_azimuth = 50 + 25 * sin(frame_counter/80);
        view_elevation = 25 + 10 * (1 - frame_counter/(frames_total/3));
    elseif frame_counter < 2*frames_total/3
        view_azimuth = 50 + 15 * cos(frame_counter/60);
        view_elevation = 25;
    else
        view_azimuth = 50 + 10 * sin(frame_counter/40);
        view_elevation = 25 + 5 * sin(frame_counter/30);
    end

    view(view_azimuth, view_elevation);

    % ========== ОБНОВЛЕНИЕ ДОПОЛНИТЕЛЬНЫХ ГРАФИКОВ ==========
    set(quality_handle, 'XData', t(1:current_idx), ...
                        'YData', Q1(1:current_idx));
    set(quality_point, 'XData', t(current_idx), ...
                       'YData', Q1(current_idx));

    set(efficiency_handle, 'XData', t(1:current_idx), ...
                           'YData', E1(1:current_idx));
    set(efficiency_point, 'XData', t(current_idx), ...
                          'YData', E1(current_idx));
```

```
    set(dV_handle, 'XData', t(1:current_idx), ...
                   'YData', dVdt(1:current_idx));
    set(dD_handle, 'XData', t(1:current_idx), ...
                   'YData', dDdt(1:current_idx));
    set(dV_point, 'XData', t(current_idx), ...
                  'YData', dVdt(current_idx));
    set(dD_point, 'XData', t(current_idx), ...
                  'YData', dDdt(current_idx));

    current_ratio = D1(current_idx) / (V1(current_idx) + eps);
    set(ratio_handle, 'XData', t(1:current_idx), ...
                      'YData', ratio(1:current_idx));
    set(ratio_point, 'XData', t(current_idx), ...
                     'YData', current_ratio);

    % ========== ОБНОВЛЕНИЕ ИНФОРМАЦИОННОЙ ПАНЕЛИ ==========
    update_info = {sprintf('СЦЕНАРИЙ 1: "БЕГ С ПРЕПЯТСТВИЯМИ"');
                   sprintf('Стратегия: Постоянная скорость, игнорирование
долга');
                   sprintf('Параметры: k = %.2f, γ = %.3f, R = %.1f', k1,
gamma1, R);
                   sprintf(' ');
                   sprintf('ТЕКУЩИЕ ПОКАЗАТЕЛИ:');
                   sprintf('  Время: %.1f недель', t(current_idx));
                   sprintf('  Ценность: %.2f у.е.', V1(current_idx));
                   sprintf('  Технический долг: %.2f у.е.', D1(current_idx));
                   sprintf('  Качество кода: %.3f', Q1(current_idx));
                   sprintf('  Эффективность: %.3f', E1(current_idx));
                   sprintf('  Отношение D/V: %.4f', current_ratio);
                   sprintf('  dV/dt: %.3f', dVdt(current_idx));
                   sprintf('  dD/dt: %.3f', dDdt(current_idx));
                   sprintf(' ');
                   sprintf('СТАТУС:');
                   sprintf('  %s', get_status_message(current_ratio,
D1(current_idx)/max(D1)))};



    % ========== СПЕЦИАЛЬНЫЕ ЭФФЕКТЫ ==========
    % Эффект при критических событиях
    for ev = 1:length(events_time)
        if abs(t(current_idx) - events_time(ev)) < 1.0
            % Вспышка
            flash_handle = scatter3(ax_main, t(current_idx),
V1(current_idx), D1(current_idx), ...
                                    300, 'o', 'MarkerFaceColor', [1 0 0], ...
                                    'MarkerEdgeColor', 'none',
'MarkerFaceAlpha', 0.3);
            pause(0.1);
            delete(flash_handle);
        end
    end

    % Эффект при пересечении критического уровня
```

```
    if current_ratio > critical_level && current_idx > 1 &&
ratio(current_idx-1) <= critical_level
        warning_text = text(ax_main, T_max/2, max(V1)*0.7, max(D1)*0.8, ...
                            '⚠️ КРИТИЧЕСКИЙ УРОВЕНЬ ДОЛГА ДОСТИГНУТ! ⚠️', ...
                            'FontSize', 14, 'Color', [1 0.2 0.2], ...
                            'FontWeight', 'bold', 'HorizontalAlignment',
'center');
        drawnow;
        pause(1);
        delete(warning_text);
    end

    % ========== УПРАВЛЕНИЕ СКОРОСТЬЮ АНИМАЦИИ ==========
    drawnow;

    % Динамическая пауза для плавности
    if current_idx < length(t)/4
        pause(0.04);
    elseif current_idx < length(t)/2
        pause(0.03);
    elseif current_idx < 3*length(t)/4
        pause(0.02);
    else
        pause(0.01);
    end

    % ========== СОХРАНЕНИЕ КАДРОВ ДЛЯ GIF ==========
    % Раскомментировать для создания анимации
    % if mod(frame_counter, 3) == 0
    %     frame = getframe(main_fig);
    %     im = frame2im(frame);
    %     [imind, cm] = rgb2ind(im, 256);
    %
    %     if frame_counter == 3
    %         imwrite(imind, cm, 'scenario1_animation.gif', 'gif', ...
    %                 'Loopcount', inf, 'DelayTime', 0.05);
    %     else
    %         imwrite(imind, cm, 'scenario1_animation.gif', 'gif', ...
    %                 'WriteMode', 'append', 'DelayTime', 0.05);
    %     end
    % end
end
```

Запуск продвинутой анимации...

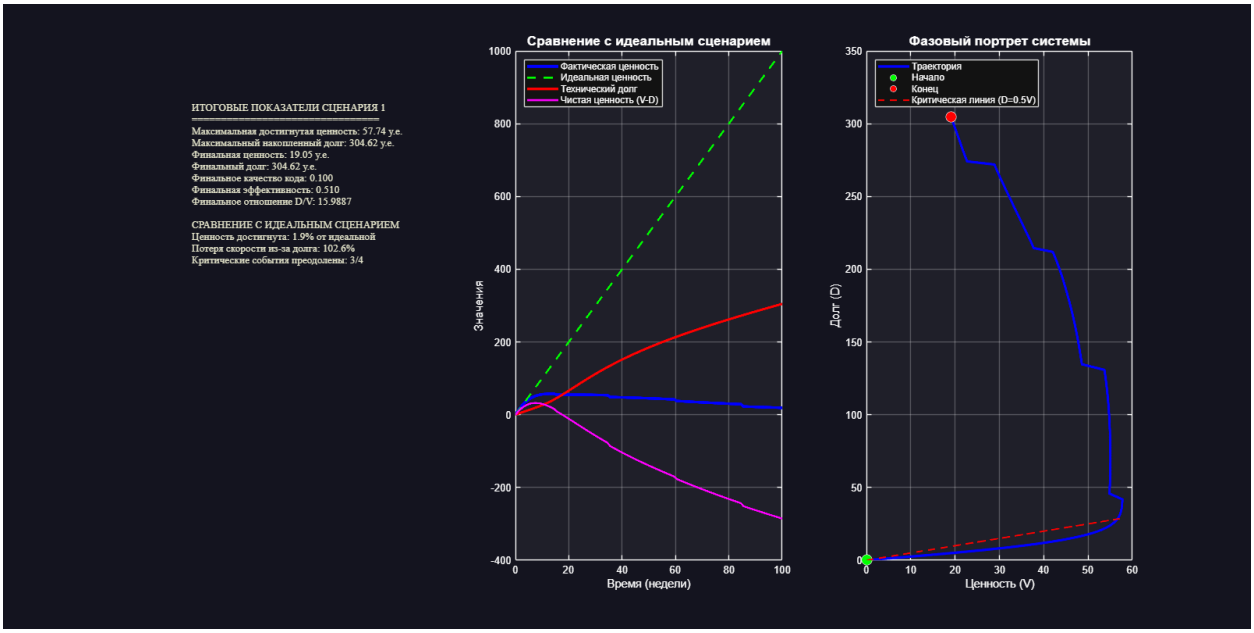


========== ФИНАЛЬНАЯ ВИЗУАЛИЗАЦИЯ ==========

```
fprintf('\nАнимация завершена!\n');

% Создание итоговой панели анализа
final_fig = figure('Position', [100, 100, 1600, 800], ...
                   'Color', [0.08 0.08 0.12], ...
                   'Name', 'Итоговый анализ сценария 1');

% Итоговые показатели
final_stats = {sprintf('ИТОГОВЫЕ ПОКАЗАТЕЛИ СЦЕНАРИЯ 1');
               sprintf('================================');
               sprintf('Максимальная достигнутая ценность: %.2f у.е.',
max(V1));
               sprintf('Максимальный накопленный долг: %.2f у.е.', max(D1));
               sprintf('Финальная ценность: %.2f у.е.', V1(end));
               sprintf('Финальный долг: %.2f у.е.', D1(end));
               sprintf('Финальное качество кода: %.3f', Q1(end));
               sprintf('Финальная эффективность: %.3f', E1(end));
               sprintf('Финальное отношение D/V: %.4f', ratio(end));
               sprintf(' ');
               sprintf('СРАВНЕНИЕ С ИДЕАЛЬНЫМ СЦЕНАРИЕМ');
               sprintf('Ценность достигнута: %.1f%% от идеальной', V1(end)/
ideal_V(end)*100);
               sprintf('Потеря скорости из-за долга: %.1f%%', (1 - dVdt(end)/
R)*100);
               sprintf('Критические события преодолены: %d/%d', ...
                       sum(D1(events_time/dt) < 0.7*max(D1)),
length(events_time))};
```

```
subplot(1,3,1);
axis off;
text(0.1, 0.9, final_stats, 'FontSize', 10, 'Color', [0.9 0.9 0.8], ...
     'VerticalAlignment', 'top', 'FontName', 'Monospaced');

% График сравнения с идеальным сценарием
subplot(1,3,2);
plot(t, V1, 'b-', 'LineWidth', 3);
hold on;
plot(t, ideal_V, 'g--', 'LineWidth', 2);
plot(t, D1, 'r-', 'LineWidth', 2.5);
plot(t, V1 - D1, 'm-', 'LineWidth', 2);

grid on;
xlabel('Время (недели)', 'FontSize', 11, 'Color', 'white');
ylabel('Значения', 'FontSize', 11, 'Color', 'white');
title('Сравнение с идеальным сценарием', 'FontSize', 13, 'Color', 'white',
'FontWeight', 'bold');
legend({'Фактическая ценность', 'Идеальная ценность', 'Технический долг',
'Чистая ценность (V-D)'}, ...
       'TextColor', 'white', 'Location', 'northwest');
set(gca, 'Color', [0.12 0.12 0.16], 'XColor', 'white', 'YColor', 'white');

% Фазовый портрет
subplot(1,3,3);
plot(V1, D1, 'b-', 'LineWidth', 2.5);
hold on;
scatter(V1(1), D1(1), 100, 'go', 'filled', 'MarkerEdgeColor', 'white');
scatter(V1(end), D1(end), 100, 'ro', 'filled', 'MarkerEdgeColor', 'white');

% Критические линии
critical_D_V = 0.5 * V1;
plot(V1, critical_D_V, 'r--', 'LineWidth', 1.5);

grid on;
xlabel('Ценность (V)', 'FontSize', 11, 'Color', 'white');
ylabel('Долг (D)', 'FontSize', 11, 'Color', 'white');
title('Фазовый портрет системы', 'FontSize', 13, 'Color', 'white',
'FontWeight', 'bold');
legend({'Траектория', 'Начало', 'Конец', 'Критическая линия (D=0.5V)'}, ...
       'TextColor', 'white', 'Location', 'northwest');
set(gca, 'Color', [0.12 0.12 0.16], 'XColor', 'white', 'YColor', 'white');

% Финальный вывод
fprintf('\n================================================\n');
fprintf('АНАЛИЗ СЦЕНАРИЯ 1 ЗАВЕРШЕН\n');
fprintf('================================================\n');
fprintf('Выводы:\n');
fprintf('  1. Начальный быстрый рост ценности\n');
fprintf('  2. Постепенное замедление из-за накопления долга\n');
fprintf('  3. Критические события усиливают негативные эффекты\n');
fprintf('  4. Система приближается к "стене" технического долга\n');
fprintf('  5. Финальная эффективность значительно ниже начальной\n');
```

```
Анимация завершена!

================================================
АНАЛИЗ СЦЕНАРИЯ 1 ЗАВЕРШЕН
================================================
Выводы:
  1. Начальный быстрый рост ценности
  2. Постепенное замедление из-за накопления долга
  3. Критические события усиливают негативные эффекты
  4. Система приближается к "стене" технического долга
  5. Финальная эффективность значительно ниже начальной
```



========== ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ ==========

```
function status_msg = get_status_message(ratio, debt_level)
    if ratio > 0.7
        status_msg = 'КРИТИЧЕСКИЙ - Система близка к коллапсу';
    elseif ratio > 0.5
        status_msg = 'ОПАСНЫЙ - Требуется немедленное вмешательство';
    elseif ratio > 0.3
        status_msg = 'РИСКОВАННЫЙ - Наблюдается замедление развития';
    elseif debt_level > 0.8
        status_msg = 'ВЫСОКАЯ НАГРУЗКА - Долг приближается к максимуму';
    elseif debt_level > 0.6
        status_msg = 'УМЕРЕННЫЙ - Долг влияет на производительность';
    else
        status_msg = 'СТАБИЛЬНЫЙ - Система развивается нормально';
    end
end
```

Published with MATLAB® R2025b